



日本国特許庁
PATENT OFFICE
JAPANESE GOVERNMENT

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日

Date of Application:

2000年10月23日

出願番号

Application Number:

特願2000-323178

出願人

Applicant (s):

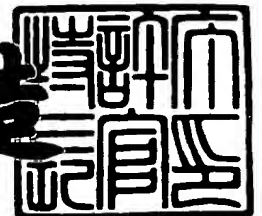
株式会社日立製作所

株式会社日立超エル・エス・アイ・システムズ

2001年 1月12日

特許庁長官
Commissioner,
Patent Office

及川耕造



出証番号 出証特2000-3111666

【書類名】 特許願

【整理番号】 H00014211

【提出日】 平成12年10月23日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 12/14

【発明者】

【住所又は居所】 東京都小平市上水本町五丁目20番1号 株式会社 日立製作所 半導体グループ内

【氏名】 谷本 千晶

【発明者】

【住所又は居所】 東京都小平市上水本町五丁目20番1号 株式会社 日立製作所 半導体グループ内

【氏名】 中田 邦彦

【発明者】

【住所又は居所】 東京都小平市上水本町5丁目22番1号 日立超エル・エス・アイ・システムズ内

【氏名】 成吉 雄一郎

【発明者】

【住所又は居所】 東京都小平市上水本町5丁目22番1号 日立超エル・エス・アイ・システムズ内

【氏名】 塚元 卓

【発明者】

【住所又は居所】 東京都小平市上水本町5丁目22番1号 日立超エル・エス・アイ・システムズ内

【氏名】 平林 茂雄

【発明者】

【住所又は居所】 東京都小平市上水本町5丁目22番1号 日立超エル・エス・アイ・システムズ内

【氏名】 渡瀬 弘

【発明者】

【住所又は居所】 東京都小平市上水本町5丁目22番1号 日立超エル・
エス・アイ・システムズ内

【氏名】 ▲高▼橋 雅聡

【特許出願人】

【識別番号】 000005108

【氏名又は名称】 株式会社 日立製作所

【特許出願人】

【識別番号】 000233169

【氏名又は名称】 株式会社 日立超エル・エス・アイ・システムズ

【代理人】

【識別番号】 100081938

【弁理士】

【氏名又は名称】 徳若 光政

【電話番号】 0422-46-5761

【先の出願に基づく優先権主張】

【出願番号】 特願2000- 3297

【出願日】 平成12年 1月12日

【手数料の表示】

【予納台帳番号】 000376

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9003106

【包括委任状番号】 9107732

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 ICカードとマイクロコンピュータ

【特許請求の範囲】

【請求項 1】 外部端子がリードライト装置と電氣的に接続されることによって動作電圧が供給され、かつ、暗号化処理又は復号化処理を伴ったデータの入出力動作を含む ICカードであって、

上記暗号化処理又は復号化処理に攪乱目的のダミー処理動作を含ませて内部回路の動作タイミング及び動作電流の画一化を行なうことを特徴とする ICカード

【請求項 2】 請求項 1 において、

上記暗号化処理又は復号化処理は、RSA暗号法などに応用可能なべき乗剰余乗算動作を含むものであることを特徴とする ICカード。

【請求項 3】 請求項 2 において、

上記べき乗剰余乗算動作は、中央処理装置からの指示を受けて動作する暗号処理用演算ユニットにより行なわれるものであることを特徴とする ICカード。

【請求項 4】 請求項 3 において、

上記暗号化処理用演算ユニットは、入力された X 、 Y 及び N を受け、 $A = 1$ 、 $B = X$ として、 $A = A^2 \bmod N$ と $A = AB \bmod N$ の演算を交互に行ない、かかる演算において Y の上位から 1 ビットずつみて、論理 0 であれば上記 $A^2 \bmod N$ の演算結果を有効なデータとして記憶回路に取り込み、論理 1 であれば上記 $A^2 \bmod N$ と $AB \bmod N$ の演算結果を有効なデータとして記憶回路に取り込むものであり、上記論理 0 のときの $A = AB \bmod N$ の演算動作が上記攪乱目的のダミー処理動作とされることを特徴とする ICカード。

【請求項 5】 請求項 4 において、

上記記憶回路は、リードライトバッファとかかるリードライトバッファを通じてデータの入出力が行なわれる複数のレジスタとからなるレジスタブロックであり、

上記演算結果は、上記 Y の特定ビット e_i の論理 1 又は 0 によってゲート回路を制御し、所定のレジスタに供給されるライトストロブ信号の伝達を制御して

、有効なデータのみがリードライトバッファを通して上記所定のレジスタに格納されることを特徴とする IC カード。

【請求項 6】 請求項 4 において、

上記記憶回路は、リードライトバッファとかかるリードライトバッファを通してデータの入出力が行なわれる複数のレジスタとからなるレジスタブロックであり、

上記演算結果は、上記 Y の特定ビット e_i の論理 1 又は 0 によってゲート回路を制御し、上記リードライトバッファに供給されるライトストロブ信号の伝達を制御して、有効なデータのみがリードライトバッファを通して上記所定のレジスタに格納されることを特徴とする IC カード。

【請求項 7】 請求項 4 において、

上記記憶回路は、リードライトバッファとかかるリードライトバッファを通してデータの入出力が行なわれる複数のレジスタ及びダミーレジスタとからなるレジスタブロックであり、

上記演算結果は、上記リードライトバッファと上記ダミーレジスタ及び複数のレジスタとの間に設けられたセクタを上記 Y の特定ビット e_i の論理 1 又は 0 によって制御して上記リードライトバッファに書き込まれた演算結果のうち有効なデータが所定のレジスタに格納され、無効なデータが上記ダミーレジスタに格納されるものであることを特徴とする IC カード。

【請求項 8】 請求項 3 において、

上記暗号化処理用演算ユニットは、入力された X、Y 及び N を受け、 $A = 1$ 、 $B = X$ として、 $A = A^2 \bmod N$ と $A = AB \bmod N$ の演算を交互に行ない、かかる演算において Y の上位から 1 ビットずつみて、論理 0 であれば上記 $A^2 \bmod N$ の演算結果を有効なデータとしてその出力タイミングで記憶回路に取り込み、論理 1 であれば上記 $A^2 \bmod N$ と $AB \bmod N$ の演算結果を有効なデータとしてその出力タイミングで記憶回路に取り込むものであり、上記 $A = A^2 \bmod N$ の演算結果が出力されてから上記 $A = AB \bmod N$ の演算が開始されるまでの間も上記 $A = A^2 \bmod N$ の動作を継続し、 $A = AB \bmod N$ の演算結果が出力されてから Y のビットの変更判定処理を含めて次のビットに対応した $A^2 \bmod$

Nの演算が開始されるまでの間も上記 $A = AB \bmod N$ の動作を継続するものであることを特徴とするICカード。

【請求項 9】 請求項 3 において、

上記暗号化処理用演算ユニットは、入力されたX、Y及びNを受け、 $A = 1$ 、 $B = X$ として、 $A = A^2 \bmod N$ と $A = AB \bmod N$ の演算とそれぞれに対してオーバーフロー演算行ない、かかる演算においてYの上位から1ビットずつみて、論理0であれば上記 $A^2 \bmod N$ の演算結果を有効なデータとして記憶回路に取り込み、論理1であれば上記 $A^2 \bmod N$ と $AB \bmod N$ の演算結果を有効なデータとして記憶回路に取り込むものであり、上記論理0のときの $A = AB \bmod N$ の演算動作と、各演算動作での不要なオーバーフロー演算が上記攪乱目的のダミー処理動作とされることを特徴とするICカード。

【請求項 10】 外部端子がリードライト装置と電氣的に接続されることによって動作電圧が供給され、かつ、暗号化処理又は復号化処理を伴ってデータの入出力動作が行われるICカードであって、

上記暗号化処理又は復号化処理に攪乱目的のダミー演算を含ませて内部回路の動作タイミング及び動作電流に不規則性を持たせてなることを特徴とするICカード。

【請求項 11】 外部端子がリードライト装置と電氣的に接続されることによって動作電圧が供給され、かつ、暗号化処理又は復号化処理を伴ってデータの入出力動作が行われるICカードであって、

上記暗号化処理又は復号化処理における各演算の間隔に攪乱目的のダミーサイクルを含ませて内部回路の動作タイミング及び動作電流に不規則性を持たせてなることを特徴とするICカード。

【請求項 12】 暗号化処理又は復号化処理を伴ったデータの入出力動作を含むモジュール構成のマイクロコンピュータであって、

上記暗号化処理又は復号化処理に攪乱目的のダミー処理動作を含ませて内部回路の動作タイミング及び動作電流の画一化を行なうことを特徴とするマイクロコンピュータ。

【請求項 13】 請求項 12 において、

上記モジュール構成は、1つの半導体基板上において形成されることによって実現されることを特徴とするマイクロコンピュータ。

【請求項14】 請求項13において、

上記暗号化処理又は復号化処理は、RSA暗号法などに応用可能なべき乗剰余乗算動作を含み、

上記べき乗剰余乗算動作は、中央処理装置からの指示を受けて動作する暗号処理用演算ユニットにより行なわれるものであることを特徴とするマイクロコンピュータ。

【請求項15】 請求項14において、

上記暗号化処理用演算ユニットは、入力されたX、Y及びNを受け、 $A = 1$ 、 $B = X$ として、 $A = A^2 \bmod N$ と $A = AB \bmod N$ の演算を交互に行ない、かかる演算においてYの上位から1ビットずつみて、論理0であれば上記 $A^2 \bmod N$ の演算結果を有効なデータとして記憶回路に取り込み、論理1であれば上記 $A^2 \bmod N$ と $AB \bmod N$ の演算結果を有効なデータとして記憶回路に取り込むものであり、上記論理0のときの $A = AB \bmod N$ の演算動作が上記攪乱目的のダミー処理動作とされることを特徴とするマイクロコンピュータ。

【請求項16】 請求項14において、

上記暗号化処理用演算ユニットは、入力されたX、Y及びNを受け、 $A = 1$ 、 $B = X$ として、 $A = A^2 \bmod N$ と $A = AB \bmod N$ の演算を交互に行ない、かかる演算においてYの上位から1ビットずつみて、論理0であれば上記 $A^2 \bmod N$ の演算結果を有効なデータとしてその出力タイミングで記憶回路に取り込み、論理1であれば上記 $A^2 \bmod N$ と $AB \bmod N$ の演算結果を有効なデータとしてその出力タイミングで記憶回路に取り込むものであり、上記 $A = A^2 \bmod N$ の演算結果が出力されてから上記 $A = AB \bmod N$ の演算が開始されるまでの間も上記 $A = A^2 \bmod N$ の動作を継続し、 $A = AB \bmod N$ の演算結果が出力されてからYのビットの変更判定処理を含めて次のビットに対応した $A^2 \bmod N$ の演算が開始されるまでの間も上記 $A = AB \bmod N$ の動作を継続するものであることを特徴とするマイクロコンピュータ。

【請求項17】 請求項14において、

上記暗号化処理用演算ユニットは、入力された X 、 Y 及び N を受け、 $A = 1$ 、 $B = X$ として、 $A = A^2 \bmod N$ と $A = AB \bmod N$ の演算とそれぞれに対してオーバーフロー演算行ない、かかる演算において Y の上位から 1 ビットずつみて、論理 0 であれば上記 $A^2 \bmod N$ の演算結果を有効なデータとして記憶回路に取り込み、論理 1 であれば上記 $A^2 \bmod N$ と $AB \bmod N$ の演算結果を有効なデータとして記憶回路に取り込むものであり、上記論理 0 のときの $A = AB \bmod N$ の演算動作と、各演算動作での不要なオーバーフロー演算が上記攪乱目的のダミー処理動作とされることを特徴とするマイクロコンピュータ。

【請求項 1 8】 請求項 3 において、

上記暗号化処理用演算ユニットは、入力された X 、 Y 及び N を受け、 $A = 1$ 、 $B = X$ として、 Y のビットの値に応じて、 $A = A^2 R^{-1} \bmod N$ 、 $A = AB R^{-1} \bmod N$ の演算を行うとともに、

演算結果にオーバーフローが発生した場合にはさらに上記演算結果 W から N の減算 $W - N$ を行なう正規動作と、各々の演算結果にオーバーフローが発生しない場合でも上記減算 $W - N$ に対応した演算による無効データを生成する攪乱目的のダミー動作を行い、

上記オーバーフローの有無に対応して有効なデータを出力させることを特徴とする IC カード。

【請求項 1 9】 請求項 1 8 において、

上記 $A^2 R^{-1} \bmod N$ 又は $AB R^{-1} \bmod N$ の演算結果 W は第 1 の記憶回路に格納され、

演算器のオーバーフローフラグ OV の有無が記憶され、

上記剰余乗算の後に上記第 1 記憶回路の演算結果 W から N の減算 $W - N$ が行われて、その演算結果が上記オーバーフローフラグ OV が有る時には上記第 1 の記憶回路に格納され、オーバーフローフラグ OV が無い時には上記第 1 記憶回路とは異なる第 2 の記憶回路に上記錯乱目的のダミー動作として書き込まれ、

上記第 1 の記憶回路の演算結果が有効なデータとして出力されることを特徴とする IC カード。

【請求項 2 0】 請求項 1 8 において、

上記 $A^2 R^{-1} \bmod N$ 又は $ABR^{-1} \bmod N$ の演算結果 W は第 1 の記憶回路に格納され、

演算器のオーバーフローフラグ OV の有無が記憶され、

上記剰余乗算の後に上記第 1 の記憶回路の演算結果 W から N の減算 $W - N$ が行われて、オーバーフローフラグ OV が有るときに上記演算結果 $W - N$ がセレクトタにより選択され、オーバーフローフラグ OV が無いときには上記第 1 記憶回路の演算結果 W がセレクトタにより選択されて第 2 の記憶回路に格納されて有効なデータとして出力されること特徴とする IC カード。

【請求項 2 1】 請求項 1 8 において、

上記 $A^2 R^{-1} \bmod N$ 又は $ABR^{-1} \bmod N$ の演算結果 W は第 1 の記憶回路に格納され、

演算器のオーバーフローフラグ OV の有無が記憶され、

上記剰余乗算の後に上記第 1 の記憶回路の演算結果 W から N の減算 $W - N$ が行われ、オーバーフローフラグ OV が有るときには減算 $W - N$ が第 2 の記憶回路に記憶され、オーバーフローフラグ OV が無いときには減算 $W - N$ が第 3 の記憶回路に記憶され、

オーバーフローフラグ OV が有るときには上記第 2 の記憶回路のデータが有効なデータとして出力され、

オーバーフローフラグ OV が無いときには上記第 1 の記憶回路のデータが有効なデータとして出力されること特徴とする IC カード。

【請求項 2 2】 請求項 1 8 において、

上記 $A^2 R^{-1} \bmod N$ 又は $ABR^{-1} \bmod N$ の演算結果 W は第 1 の記憶回路に格納され、

演算器のオーバーフローフラグ OV の有無が記憶され、

上記剰余乗算の後に上記第 1 の記憶回路の演算結果 W から N の減算結果 $W - N$ が第 2 の記憶回路に格納され、オーバーフローフラグ OV が無いとき第 1 の記憶回路と第 2 の記憶回路を選択する最下位アドレスを逆転させて、上記第 2 の記憶回路を選択するアドレスにより第 1 の記憶回路を選択して有効なデータとして出力させ、オーバーフローフラグ OV が有るとき第 1 の記憶回路と第 2 の記憶回路

を選択する最下位アドレスをそのままにして第 2 の記憶回路の演算結果を有効なデータとして出力させることを特徴とする IC カード。

【請求項 2 3】 請求項 1 8 において、

上記 $A^2 R^{-1} \bmod N$ 又は $ABR^{-1} \bmod N$ の演算結果 W は第 1 の記憶回路に格納され、

演算器のオーバーフローフラグ OV の有無が記憶され、

上記剰余乗算の後に上記第 1 の記憶回路と第 2 の記憶回路のアドレスが交換され、第 2 の記憶回路を選択するアドレスにより選択される演算結果値 W から N の減算 $W - N$ が行われて第 1 の記憶回路を選択するアドレスにより選択される第 2 の記憶回路に減算結果 $W - N$ が格納され、オーバーフローフラグ OV が有るときにのみ上記アドレスを再度交換し、第 1 の記憶回路を選択するアドレスにより選択される第 1 又は第 2 の記憶回路のデータを有効なデータとして出力させることを特徴とする IC カード。

【請求項 2 4】 請求項 1 8 において、

上記 $A^2 R^{-1} \bmod N$ 又は $ABR^{-1} \bmod N$ の演算結果 W は第 1 の記憶回路に格納され、

上記剰余乗算の後に上記第 1 の記憶回路の演算結果値 W から N の減算 $W - N$ が行われて第 2 の記憶回路に格納され、

この $W - N$ の減算が行われた時の演算器からボロフラグ BR が記憶され、

ボロフラグ BR が有るときには、第 1 の記憶回路と第 2 の記憶回路を選択する最下位アドレスを逆転させて、上記第 2 の記憶回路を選択するアドレスにより第 1 の記憶回路の演算結果 W を出力し、

ボロフラグ BR が無いときには、第 1 の記憶回路と第 2 の記憶回路を選択する最下位アドレスをそのままして、上記第 2 の記憶回路を選択するアドレスにより第 2 の記憶回路の演算結果 $W - N$ を出力させることを特徴とする IC カード。

【請求項 2 5】 請求項 1 4 において、

上記暗号化処理用演算ユニットは、入力された X 、 Y 及び N を受け、 $A = 1$ 、 $B = X$ として、 Y のビットの値に応じて、 $A = A^2 R^{-1} \bmod N$ 、 $A = ABR^{-1} \bmod N$ の演算を行うとともに、

演算結果にオーバーフローが発生した場合にはさらに上記演算結果WからNの減算 $W - N$ を行なう正規動作と、各々の演算結果にオーバーフローが発生しない場合でも上記減算 $W - N$ に対応した演算による無効データを生成する攪乱目的のダミー動作を行い、

上記オーバーフローの有無に対応して有効なデータを出力させることを特徴とするマイクロコンピュータ。

【請求項 2 6】 請求項 2 5 において、

上記 $A^2 R^{-1} \bmod N$ 又は $ABR^{-1} \bmod N$ の演算結果Wは第 1 の記憶回路に格納され、

演算器からのオーバーフローフラグOVの有無が記憶され、

上記剰余乗算の後に上記第 1 記憶回路の演算結果WからNの減算 $W - N$ が行われて、その演算結果が上記オーバーフローフラグOVが有る時には上記第 1 の記憶回路に格納され、オーバーフローフラグOVが無い時には上記第 1 記憶回路とは異なる第 2 の記憶回路に上記錯乱目的のダミー動作として書き込まれ、

上記第 1 の記憶回路の演算結果が有効なデータとして出力されることを特徴とするマイクロコンピュータ。

【請求項 2 7】 請求項 2 5 において、

上記 $A^2 R^{-1} \bmod N$ 又は $ABR^{-1} \bmod N$ の演算結果Wは第 1 の記憶回路に格納され、

演算器のオーバーフローフラグOVの有無が記憶され、

上記剰余乗算の後に上記第 1 の記憶回路の演算結果WからNの減算 $W - N$ が行われて、オーバーフローフラグOVが有るときに上記演算結果 $W - N$ がセレクタにより選択され、オーバーフローフラグOVが無いときには上記第 1 記憶回路の演算結果Wがセレクタにより選択されて第 2 の記憶回路に格納されて有効なデータとして出力されること特徴とするマイクロコンピュータ。

【請求項 2 8】 請求項 2 5 において、

上記 $A^2 R^{-1} \bmod N$ 又は $ABR^{-1} \bmod N$ の演算結果Wは第 1 の記憶回路に格納され、

演算器のオーバーフローフラグOVの有無が記憶され、

上記剰余乗算の後に上記第 1 の記憶回路の演算結果 W から N の減算 $W - N$ が行われ、オーバーフローフラグ OV が有るときには減算結果 $W - N$ が第 2 の記憶回路に記憶され、オーバーフローフラグ OV が無いときには減算結果 $W - N$ が第 3 の記憶回路に記憶され、

オーバーフローフラグ OV が有るときには上記第 2 の記憶回路のデータが有効なデータとして出力され、

オーバーフローフラグ OV が無いときには上記第 1 の記憶回路のデータが有効なデータとして出力されること特徴とするマイクロコンピュータ。

【請求項 2 9】 請求項 2 5 において、

上記 $A^2 R^{-1} \bmod N$ 又は $ABR^{-1} \bmod N$ の演算結果 W は第 1 の記憶回路に格納され、

演算器のオーバーフローフラグ OV の有無が記憶され、

上記剰余乗算の後に上記第 1 の記憶回路の演算結果 W から N の減算結果 $W - N$ が第 2 の記憶回路に格納され、オーバーフローフラグ OV が無いとき第 1 の記憶回路と第 2 の記憶回路を選択する最下位アドレスを逆転させて、上記第 2 の記憶回路を選択するアドレスにより第 1 の記憶回路を選択して有効なデータとして出力させ、オーバーフローフラグ OV が有るとき第 1 の記憶回路と第 2 の記憶回路を選択する最下位アドレスをそのままにして第 2 の記憶回路の演算結果を有効なデータとして出力させることを特徴とするマイクロコンピュータ。

【請求項 3 0】 請求項 2 5 において、

上記 $A^2 R^{-1} \bmod N$ 又は $ABR^{-1} \bmod N$ の演算結果 W は第 1 の記憶回路に格納され、

演算器のオーバーフローフラグ OV の有無が記憶され、

上記剰余乗算の後に上記第 1 の記憶回路と第 2 の記憶回路のアドレスが交換され、第 2 の記憶回路を選択するアドレスにより選択される演算結果値 W から N の減算 $W - N$ が行われて第 1 の記憶回路を選択するアドレスにより選択される第 2 の記憶回路に減算結果 $W - N$ が格納され、オーバーフローフラグ OV が有るときにのみ上記アドレスを再度交換し、第 1 の記憶回路を選択するアドレスにより選択される第 1 又は第 2 の記憶回路のデータを有効なデータとして出力させること

を特徴とするマイクロコンピュータ。

【請求項 3 1】 請求項 2 5 において、

上記 $A^2 R^{-1} \bmod N$ 又は $ABR^{-1} \bmod N$ の演算結果 W は第 1 の記憶回路に格納され、

上記剰余乗算の後に上記第 1 の記憶回路の演算結果値 W から N の減算 $W - N$ が行われて第 2 の記憶回路に格納され、

この $W - N$ の減算が行われた時の演算器からボロフラグ BR が記憶され、

ボロフラグ BR が有るときには、第 1 の記憶回路と第 2 の記憶回路を選択する最下位アドレスを逆転させて、上記第 2 の記憶回路を選択するアドレスにより第 1 の記憶回路の演算結果 W を出力し、

ボロフラグ BR が無いときには、第 1 の記憶回路と第 2 の記憶回路を選択する最下位アドレスをそのままして、上記第 2 の記憶回路を選択するアドレスにより第 2 の記憶回路の演算結果 $W - N$ を出力させることを特徴とするマイクロコンピュータ。

【発明の詳細な説明】

【0 0 0 1】

【発明の属する技術分野】

この発明は、IC カードとマイクロコンピュータに関し、特に IC カードやプログラム内蔵の 1 チップマイクロコンピュータのような CPU とメモリを含み暗号鍵を使ったデータ処理を行なうものの機密保護技術に利用して有効な技術に関するものである。

【0 0 0 2】

【従来の技術】

メモリに保存されている鍵情報を用いてデータの暗号処理化又は復号化処理を行なうようにした IC カードにおいて、処理時間の違いを利用して実行内容や暗号鍵を推定する TA (Timing Attack) 法のようなハッキング手法に対抗するため、暗号処理化又は復号化処理の実行中又は実行の前後に、鍵情報の内容との時間的な相関関係を喪失させる遅延処理を実行する技術の例として、特開平 1 0 - 6 9 2 2 2 号がある。また、IC カードに関しては、オーム社出版電子情報通信

学会編水沢順一著「ＩＣカード」などがある。

【 0 0 0 3 】

【発明が解決しようとする課題】

近年、ＩＣカードが暗号処理を行っている時の消費電流を観測して解析することにより、容易に暗号処理の内容や暗号鍵が推定されることの可能性が示唆されている。このことについては、John Wiley & sons 社 W.Rankl & W. Effing著「Smart Card Handbook」の8.5.1.1 Passive protective mechanisms(263ページ)に記載されている。

【 0 0 0 4 】

つまり、SPA (Simple Power Analysis) 法では、演算命令の違い、あるいは処理されているデータの違いにより生じる消費電流波形の違いから、暗号鍵や処理されているデータを解析し、DPA (Differential Power Analysis) 法では、消費電流波形を統計処理して暗号鍵を推定する。このDPA法では、例えばDESのある部分に仮定した暗号鍵をあてはめて、平文を変化させながら消費電流波形を測定して統計する。暗号鍵を様々に変化させながらこの作業を繰り返し、正しい鍵のときには電流波形が大きなピークを示す。

【 0 0 0 5 】

前記公報に記載のようにTA (Timing Attack) 法のみを考慮した遅延処理では、実際の演算による消費電流の相関性までも喪失させることができず、上記のような消費電流波形を観測するというSPA又はDPA法のようなハッキング手法には対抗できない。そこで、本願発明者等においては、上記ＩＣカード及びＩＣカード等のようなモジュールに搭載されるマイクロコンピュータのように内蔵のプログラムにより一定のデータ処理動作を行うものに対して上記のような消費電流の観測による暗号処理の内容や暗号鍵の解読をより確実に防止することができる機密保護技術を開発するに至った。

【 0 0 0 6 】

この発明の目的は、機密保護の強化を実現したＩＣカードとマイクロコンピュータを提供することにある。この発明の他の目的は、機密保護のための信号処理の高速化とその強化を実現したＩＣカードとマイクロコンピュータを提供するこ

とにある。この発明の前記ならびにそのほかの目的と新規な特徴は、本明細書の記述および添付図面から明らかになるであろう。

【0007】

【課題を解決するための手段】

本願において開示される発明のうち代表的なものの概要を簡単に説明すれば、下記の通りである。すなわち、外部端子がリードライト装置と電氣的に接続されることによって動作電圧が供給され、かつ、暗号化処理又は復号化処理を伴ったデータの入出力動作を含むＩＣカードにおいて、上記暗号化処理又は復号化処理に本来の処理動作に似た攪乱目的のダミー処理動作を含ませて内部回路の動作タイミング及び動作電流の画一化を行なうようにする。

【0008】

本願において開示される発明のうち他の代表的なものの概要を簡単に説明すれば、下記の通りである。すなわち、暗号化処理又は復号化処理を伴ったデータの入出力動作を含むモジュール構成のマイクロコンピュータにおいて、上記暗号化処理又は復号化処理に本来の処理動作に似た攪乱目的のダミー処理動作を含ませて内部回路の動作タイミング及び動作電流の画一化を行なうようにする。

【0009】

【発明の実施の形態】

図１には、この発明が適用されるＩＣカードの一実施例の外観図が示されている。ＩＣカードは、プラスチックケースからなるカード１０１と、かかるカード１０１の内部に搭載された図示しない１チップのマイクロコンピュータ等からなるＩＣカード用チップを持つものである。上記ＩＣカードは、さらに上記ＩＣカード用チップの外部端子に接続されている複数の接点（電極）１０２を持つ。複数の接点１０２は、後で図２によって説明するような電源端子ＶＣＣ、電源基準電位端子ＶＳＳ、リセット入力端子ＲＥＳバー、クロック端子ＣＬＫ、データ端子Ｉ／Ｏ－１／ＩＲＱバー、Ｉ／Ｏ－２／ＩＲＱバーとされる。ＩＣカードは、かかる接点１０２を通して図示しないリーダーライタのような外部結合装置から電源供給を受け、また外部結合装置との間でのデータの通信を行う。

【0010】

図2には、この発明に係るICカードに搭載されるICカード用チップ（マイクロコンピュータ）の一実施例の概略ブロック図が示されている。同図の各回路ブロックは、公知のMOS集積回路の製造技術により、特に制限されないが、単結晶シリコンのような1個の半導体基板上において形成される。

【0011】

この発明に係るICカード用チップの構成は、基本的にマイクロコンピュータと同じような構成である。その構成は、クロック生成回路205、中央処理装置（以下単にCPUという場合がある）201、ROM(Read Only Memory)206やRAM(Random Access Memory)207、不揮発性メモリ208などの記憶装置、暗号化及び復号化処理の演算を行なうコプロセッサ209、入出力ポート（I/Oポート）202などからなる。

【0012】

クロック生成回路205は、図示しないリーダライタ（外部結合装置）から図1の接点102を介して供給される外部クロックCLKを受け、かかる外部クロック信号に同期したシステムクロック信号を形成し、それをチップ内部に供給する回路である。CPU201は、論理演算や算術演算などを行う装置であり、システムコントロールロジック、乱数発生器及びセキュリティロジック及びタイマなどを制御する。記憶装置206、207、208は、プログラムやデータを格納する装置である。コプロセッサ209は、後述するようにRSA暗号法などに応用可能なべき乗剰余乗算動作を行なう演算器とレジスタ及び制御論理から構成される。I/O（入出力）ポート202は、リーダライタと通信を行う装置である。データバス204とアドレスバス203は、各装置を相互に接続するバスである。

【0013】

上記記憶装置206、207、208のうち、ROM206は、記憶内容が不揮発的に固定されているメモリであり、主にプログラムを格納するメモリである。揮発性メモリ（以下、RAMという）207は自由に記憶情報の書き換えができるメモリであるが、電源の供給が中断されると、記憶している内容が消えてなくなる。ICカードがリーダライタから抜かれると電源の供給が中断されるため

、RAM 2 0 7の内容は、保持されなくなる。

【 0 0 1 4 】

上記不揮発性メモリ（以下、EEPROM(Electrical Erasable Programmable Read Only Memory) という）2 0 8は、内容の書き換えが可能な不揮発性メモリであり、その中に一旦書き込まれた情報は、電源の供給が停止されてもその内部に保持される。このEEPROM 2 0 8は、書き換える必要があり、かつICカードがリーダライタから抜かれても保持すべきデータを格納するために使われる。例えば、ICカードがプリペイドカードとして使用されるような場合、のプリペイドの度数などは、使用するたびに書き換えられる。この場合の度数などは、リーダライタが抜かれてもICカード内で記憶保持する必要があるため、EEPROM 2 0 8で保持される。

【 0 0 1 5 】

CPU 2 0 1は、いわゆるマイクロプロセッサと同様な構成にされる。すなわち、その詳細を図示しないけれども、その内部に命令レジスタ、命令レジスタに書込まれた命令をデコードし、各種のマイクロ命令ないしは制御信号を形成するマイクロ命令ROM、演算回路、汎用レジスタ（RG 6等）、内部バスBUSに結合するバสดライバ、バスレシーバなどの入出力回路を持つ。CPU 2 0 1は、ROM 2 0 6などに格納されている命令を読み出し、その命令に対応する動作を行う。CPU 2 0 1は、I/Oポート2 0 2を介して入力される外部データの取り込み、ROM 2 0 6からの命令や命令実行のために必要となる固定データのようなデータの読み出し、RAM 2 0 7やEEPROM 2 0 8に対するデータの書き込みと読み出し動作制御等を行う。

【 0 0 1 6 】

上記CPU 2 0 1は、クロック生成回路2 0 5から発生されるシステムクロック信号を受けそのシステムクロック信号によって決められる動作タイミング、周期をもって動作される。CPU 2 0 1は、その内部の主要部がPチャンネル型MOSFETとNチャンネル型MOSFETとからなるCMOS回路から構成される。特に制限されないが、CPU 2 0 1は、CMOSスタティックフリップフロップのようなスタティック動作可能なCMOSスタティック回路と、信号出力ノー

ドへの電荷のプリチャージと信号出力ノードへの信号出力とをシステムクロック信号に同期して行うようなCMOSダイナミック回路とを含む。

【 0 0 1 7 】

ICカードのセキュリティ機能としては、チップ内部で乱数を自動生成する乱数発生器や、ランダムに割込みを生成するタイマー機能などの他に、本願発明にかかる高セキュリティ機能として、ICカードと外部装置とのデータ送受信の際に用いるRSA暗号法などに応用可能なべき乗剰余演算動作を行なう暗号処理用演算ユニット（コプロセッサ）209を内蔵している。このコプロセッサ209は専用のレジスタが内蔵されている。

【 0 0 1 8 】

ICカードにおけるセキュリティ・システムでは、通信データの暗号処理は必須であり、この実施例でも現在最も多く利用されている公開鍵暗号としてRSA暗号が用いられる。この暗号法では、暗号化・復号化ともにべき乗剰余乗算 $X^Y \bmod N$ を用いるが、これは古くから知られている計算アルゴリズムによって剰余乗算 $A^2 \bmod N$ と $AB \bmod N$ の2つの形に分解することができる。つまり、 $Y = e_n e_{n-1} \cdots e_1$ の値 e_i を上位 e_n から最下位の e_1 まで順に1ビットずつ見ていき、 $e_i = 0$ だったら $A^2 \bmod N$ のみを、 $e_i = 1$ だったら $A^2 \bmod N$ と $AB \bmod N$ を演算する。したがって、 $e_i = 0$ のときには $A^2 \bmod N$ の演算の後に $i = 0$ であるかの判定処理が行なわれ、 $e_i = 1$ のときには $A^2 \bmod N$ と $AB \bmod N$ との演算の後に $i = 0$ であるかの判定処理が行なわれるために、 $e_i = 0$ と1とに対応した2通りの電流波形の形態が現れてしまう。

【 0 0 1 9 】

この実施例のようにコプロセッサ209を用いた場合には、その消費電流はCPUの消費電流に比べて比較的大きいため、この部分の電流波形を観測することによりコプロセッサの動作形態を比較的容易に識別することができ、前記TA法とSPA法により暗号鍵Yの値を解読されてしまう可能性が高い。そこで、この実施例のコプロセッサ209では、上記暗号化・復号化ともに用いられるべき乗剰余乗算 $X^Y \bmod N$ の演算を行なうに当たり攪乱目的のダミーの演算が挿入さ

れる。つまり、図3のタイミング図及び図4のフローチャート図に示すように $e_i = 0$ でも1でも $A^2 \bmod N$ と $AB \bmod N$ の両方の演算を常に行なうようにするものである。

【0020】

図3のタイミング図において、(a)に示すように本来は、 $e_n = 1$ のときには $A^2 \bmod N$ の演算を行い、 e_n の判定の1により時間 t_1 を経て $AB \bmod N$ の演算を行い、その演算後に i をデクリメント ($n-1$) して $i = 0$ の判定に時間 t_2 を費やす。次いで、次ビット $e_{n-1} = 0$ のときは、 $A^2 \bmod N$ の演算を行い、 $e_{n-1} = 0$ の判定と i をデクリメント ($n-2$) して $i = 0$ の判定に時間 t_3 を費やす。そして、次ビット $e_{n-2} = 1$ のときには、 $A^2 \bmod N$ の演算を行い、 e_{n-2} の判定の1により時間 t_1 を経て $AB \bmod N$ の演算を行い、その演算後に i をデクリメント ($n-3$) して $i = 0$ の判定に時間 t_2 を費やす。以下、同様に e_1 まで同様な動作を繰り返すものである。

【0021】

この実施例のコプロッサ209においては、上記暗号鍵 Y の各ビット e_i の論理0又は1に無関係に $A^2 \bmod N$ の演算の後に $AB \bmod N$ の演算を行なうようにする。図3(b)の $e_{n-1} = 0$ のときのように e_i が論理0のときにおける上記 $AB \bmod N$ の演算が攪乱目的のダミー演算として挿入される。つまり、(b)のタイミング図及び図4のフローチャート図のように、 $A^2 \bmod N$ と $AB \bmod N$ の演算動作の間には、例えば e_i の判定の判定を含む時間 t_1 が費やされ、 $AB \bmod N$ と次ビットに対応した $A^2 \bmod N$ の演算動作の間には、 i のデクリメント動作と $i = 0$ の判定時間 t_2 が費やされる画一化された動作タイミング及び動作電流とすることができる。ただし、この実施例では、 e_i の判定処理は、その結果が演算動作の分岐の条件とされないため図4のフローチャート図では省略されている。

【0022】

図5には、上記コプロセッサの一実施例のブロック図が示されている。この実施例では、主に演算器、制御論理、専用レジスタブロックより構成され、べき乗剰余演算の最終結果はデータバツファ、データバスを介して中央処理装置CPU

に送信される。専用レジスタは、アドレスバスから供給されるアドレス信号に対応してその選択動作が行なわれる。

【 0 0 2 3 】

この実施例では、内部バスMDBとレジスタブロックのリードライトバッファ (R/W Buffer)との間にゲート回路1が設けられる。このゲート回路1は、制御論理により制御が行なわれて、 e_i が論理0ならば $A^2 \bmod N$ 動作の演算結果が内部バスMDBとリードライトバッファを介して所定のレジスタCDAに取り込まれた後開いていたゲートが閉じるようにされる。つまり、上記演算結果がリードライトバッファに取り込まれると、その後にゲートを閉じてしまいリードライトバッファへの新たなデータの書き込みを禁止する。したがって、その後に行なわれる $AB \bmod N$ の演算結果は無効データとして扱われることとなる。また、 e_i が論理1ならばゲート回路1はゲートを開いた状態のままとされる。

【 0 0 2 4 】

図6には、上記コプロセッサの他の一実施例のブロック図が示されている。この実施例では、レジスタブロックのリードライトバッファ (R/W Buffer)と各レジスタとの間にゲート回路2が設けられる。このゲート回路2は、前記同様に制御論理により制御が行なわれて、 e_i が論理0ならば $A^2 \bmod N$ 動作の演算結果が内部バスMDBとリードライトバッファとを介して所定のレジスタCDAに書き込まれた後に開いていたゲートが閉じるようにされる。つまり、上記演算結果がレジスタCDAに取り込まれると、その後にゲートを閉じてしまいかかるレジスタCDAへの新たなデータの書き込みを禁止する。したがって、その後に行なわれる $AB \bmod N$ の演算結果は、リードライトバッファまでは書き込まれるが、実際には無効データとして扱われることとなる。また、 e_i が論理1ならばゲート回路2はゲートを開いた状態のままとされる。

【 0 0 2 5 】

図7には、上記ゲート回路の一実施例の内部構成図が示されている。ダミー書き込み制御ユニットは、アンドゲート回路によって構成され、一方の入力には制御論理からのライトイネーブル信号が供給され、他方の入力には演算器で生成されたライトストロブ信号が供給される。上記ゲート回路の出力信号は、データ

バッファ (R/W Buffer)と専用レジスタにライトストロブ信号として伝えられる。

【 0 0 2 6 】

この実施例では、演算結果そのものの伝達制御するものに代えて、レジスタ又はデータバッファへの書き込み動作を指示するライトストロブ信号の発生タイミングを切り換えるようにするものである。つまり、 $e_i = 0$ のときには、 $A^2 \bmod N$ 動作の演算結果が出力された後にライトイネーブル信号をロウレベルとしてアンドゲート回路のゲートが閉じるようにするものである。逆に、 $e_i = 1$ のときには、制御論理はライトイネーブル信号をハイレベルのままとして、演算器で形成されたライトストロブ信号がそのままデータバッファ又は専用レジスタに伝えられる。この構成では、複数ビットからなる演算結果 A に対応して、複数個のゲート回路を設ける必要がないので簡素化が可能になる。

【 0 0 2 7 】

図 8 には、上記コプロセッサの他の一実施例のブロック図が示されている。この実施例では、レジスタブロックのリードライトバッファ (R/W Buffer) と各レジスタとの間にセクタ 2 とレジスタブロックにダミーレジスタ 1 が設けられる。このセクタ 2 は、前記同様に制御論理により制御が行なわれて、 e_i が論理 0 ならば $A^2 \bmod N$ 動作の演算結果が内部バス MDB とリードライトバッファとを介して所定のレジスタ CDA に書き込まれるような信号経路を形成し、その後にダミーレジスタ 1 を選択するような信号経路を形成する。

【 0 0 2 8 】

つまり、上記演算結果がレジスタ CDA に取り込まれると、その後にダミーレジスタ 1 を選択するので、レジスタ CDA への新たなデータの書き込みを禁止しつつその後に行なわれる $AB \bmod N$ の演算結果がダミーレジスタに書き込まれるものとなる。 e_i が論理 1 ならばセクタ 2 は常にレジスタ CDA を選択する。この構成は、演算結果をレジスタに書き込む動作を含めて e_i が論理 0 のときと論理 1 のときとで電流波形で見たときに全く同一にすることができるから、電流波形を利用したアタックをより確実に無力化することができる。

【 0 0 2 9 】

図 9 には、この発明に係るコプロセッサの他の一実施例の動作を説明するための構成図が示されている。図 9 (a) のタイミング図及び (b) のフローチャート図において、前記説明したように、 $A^2 \bmod N$ の演算後、 e_i の判定の時間 t_1 の間もダミー演算動作として $A^2 \bmod N$ を継続して $AB \bmod N$ の演算に移行する。

【 0 0 3 0 】

その演算後に i をデクリメント (-1) して $i = 0$ の判定に時間 t_2 を費やすが、その間も上記 $AB \bmod N$ の演算を継続させる。以下、同様に e_1 まで同様な動作を繰り返すものである。この構成は、演算動作中は、 e_i が論理 0 と 1 のときに関係なく上記のような演算動作を継続するので、電流波形でみたときに格別な特徴を見出すことができないから、電流波形を利用したアタックを無力化することができる。

【 0 0 3 1 】

図 1 0 には、図 9 のコプロセッサの動作を実現するための一実施例のブロック図が示されている。制御論理では、ダミーイネーブル信号とコプロイネーブル信号を送出する。上記ダミーイネーブル信号とコプロイネーブル信号は、オアゲート回路を通して演算器に入力される。それ故、コプロイネーブル信号がアクティブであるときに加えて、ダミーイネーブル信号がアクティブであるときにも演算器は演算動作を行なうようにされる。

【 0 0 3 2 】

上記ダミーイネーブル信号は、インバータ回路を通してアンドゲート回路の一方の入力に供給され、かかるアンドゲート回路の他方の入力には演算器で形成されたライトストロープ信号が供給される。つまり、演算器で形成されたライトストロープ信号の伝達をダミーイネーブル信号で選択的に停止できるようにする。コプロイネーブル信号がアクティブにされて、前記正規の演算動作が終了すると、その演算結果を出力するためのライトストロープ信号が形成される。このようにコプロイネーブル信号がアクティブのときには、ダミーイネーブル信号の反転信号がアクティブレベルとなってアンドゲート回路のゲートを開くように制御するので、上記正規演算結果はライトストロープ信号によって、R/Wバッファ又

はレジスタブロックの所定のレジスタに書き込まれる。

【 0 0 3 3 】

上記のような正規演算が終了すると、ダミーイネーブル信号がアクティブとなって演算器に対して演算動作を指示する。この演算の終了によって、上記ライトストロブ信号が形成されるが、上記ダミーイネーブル信号の反転信号によってアンドゲート回路がゲートを閉じているので、上記攪乱目的のダミー演算動作によって発生されたライトストロブ信号がR/Wバッファ又はレジスタブロックの所定のレジスタに伝えられることはない。これにより、攪乱目的のダミー演算結果は無効データとして消失させられる。

【 0 0 3 4 】

図 1 1 には、この発明に係るコプロセッサの他の一実施例の動作を説明するためのタイミング図が示されている。前記図 3 に示した実施例のように、攪乱目的のダミー演算を挿入して、(a) のタイミング図のように、 e_i に対して画一化して $A^2 \bmod N$ と $AB \bmod N$ の演算を一对として行なうようにした場合でも、各演算には、演算結果にオーバーフロー処理を必要とするもの（あり）のものと、オーバーフロー処理を必要としないもの（なし）が発生する。

【 0 0 3 5 】

このようなオーバーフロー処理は、演算時間を長くするものであるので電流波形でみると、オーバーフロー処理ありとなしとの識別が可能になる。このような電流波形の特徴から演算内容や演算データを推測することも不可能ではないと考えられるため、この実施例では (b) のタイミング図に示すようにオーバーフロー処理を不要とする演算に対しても必要なときと同様にオーバーフロー処理を挿入する。つまり、みかけ上は、全ての演算 $A^2 \bmod N$ と $AB \bmod N$ の演算において画一的にオーバーフロー処理のための動作が実施されるために、その識別を無力化するものである。

【 0 0 3 6 】

図 1 2 は、この発明に係るコプロセッサの他の一実施例の動作を説明するためのフローチャート図が示されている。このフローチャート図は、前記図 1 1 (b) に対応している。 $A^2 \bmod N$ と $AB \bmod N$ の各演算は、剰余演算部とオー

バーフロー演算部からなり、演算結果に無関係に上記オーバーフロー演算処理を実施するものである。

【0037】

図13には、この発明に係るコプロセッサの他の一実施例の動作の詳細を説明するためのタイミング図が示されている。この実施例による対策前では、前記 $A^2 \bmod N$ と $AB \bmod N$ のようなコプロ演算においては、その演算結果に対応してオーバーフロー処理のあるものと無いもの2種類が存在したが、この実施例による対策後では、前記 $A^2 \bmod N$ と $AB \bmod N$ のようなコプロ演算においては、その演算結果に無関係に常にオーバーフロー処理が実行される。このため、本来はオーバーフロー処理が不要な演算動作に対して実施されたオーバーフロー処理は、攪乱目的のダミー動作とされる。

【0038】

図14には、図11ないし図13に示したコプロセッサの動作を実現するための一実施例のブロック図が示されている。制御論理では、ダミーオーバーフロー信号とコプロオーバーフロー信号を送出する。上記ダミーオーバーフロー信号とコプロオーバーフロー信号は、オアゲート回路を通して演算器に入力される。それ故、コプロオーバーフロー信号がアクティブであるときに加えて、ダミーオーバーフロー信号がアクティブであるときにも演算器はオーバーフロー処理動作を行なうようにされる。

【0039】

上記コプロオーバーフロー信号は、アンドゲート回路の一方の入力に供給され、かかるアンドゲート回路の他方の入力に演算器で形成されたライトストロープ信号が供給される。つまり、演算器で形成されたライトストロープ信号の伝達をコプロオーバーフロー信号がアクティブレベルでないときに選択的に停止できるようにする。つまり、コプロオーバーフロー信号がアクティブレベルでないときはダミーオーバーフロー信号によって演算器がオーバーフロー処理を行なっているので、かかるオーバーフロー処理で形成されたライトストロープ信号は上記ゲート回路のゲートを閉じることによって無効にするものである。したがって、前記正規のオーバーフロー処理終了すると、その処理結果を出力するためのライト

ストロブ信号が形成されて、R/Wバッファ又はレジスタブロックの所定のレジスタに処理結果が書き込まれる。

【 0 0 4 0 】

これに対して、ダミーオーバーフロー信号がアクティブとなって演算器に対してオーバーフロー処理動作を指示した場合には、そのオーバーフロー処理によって形成されたライトストロブ信号は、上記コプロオーバーフロー信号によってアンドゲート回路のゲートが閉じられるものであるから、上記攪乱目的のダミーオーバーフロー処理動作によって発生されたライトストロブ信号がR/Wバッファ又はレジスタブロックの所定のレジスタに伝えられることはない。これにより、攪乱目的のダミーオーバーフロー処理結果は無効データとして消失させられる。

【 0 0 4 1 】

図 1 5 には、この発明に係るコプロセッサの更に他の一実施例の動作を説明するためのタイミング図が示されている。(a)に示すように本来は、 $e_n = 1$ のときには $A^2 \bmod N$ の演算を行い、 e_n の判定の1により時間 t_1 を経て $AB \bmod N$ の演算を行い、その演算後に i をデクリメント($n-1$)して $i=0$ の判定に時間 t_2 を費やす。次いで、次ビット $e_{n-1} = 0$ のときは、 $A^2 \bmod N$ の演算を行い、 $e_{n-1} = 0$ の判定と i をデクリメント($n-2$)して $i=0$ の判定に時間 t_3 を費やすような演算動作に対して、上記各演算毎の時間 t_1 、 t_2 及び t_3 に対して攪乱目的のダミーサイクルが挿入される。

【 0 0 4 2 】

(b)のタイミング図では、上記攪乱目的のダミーサイクルの挿入は、各演算毎の時間を最も長い時間 t_3 に揃えるように挿入するものである。これにより、時間 t_3 をインターバルとして $A^2 \bmod N$ 又は $AB \bmod N$ のいずれかの演算が実施されるために、みかけ上は演算動作に対応した電流波形が画一化されてその識別を無力化するものである。これに対して、(c)タイミング図では、上記(b)とは逆に上記演算毎のインターバルにおいて時間がランダムに変化する攪乱目的のダミーサイクルが挿入される。上記 $A^2 \bmod N$ 又は $AB \bmod N$ のいずれかの演算が時間的にランダムに実施される。そのため、電流波形でみると上

記各演算動作と無関係で、かつ不規則性の電流値にされる。言い換えるならば、上記演算器において同じ状態及び同じ動作でも毎回異なるよう、統計的な観点での非再現性を持つようにされるために、その識別を無力化することができる。

【 0 0 4 3 】

上記のような攪乱目的のダミーサイクルは、前記図 2 に示されたようにタイマーを利用して演算間隔を可変にするものである。あるいはコプロセッサの外部にタイマーを設けて一定の時間が経過するまで次の演算の実行を待つようにする。つまり、コプロセッサによるべき乗剰余乗算の演算において、図 1 5 (a) に示した前記各演算毎の時間 t_1 , t_2 , t_3 に攪乱目的のダミーのサイクルを挿入し、一定時間後にタイマーからの割込みを入れる。これにより、図 1 5 (b) に示すように t_1 , t_2 , t_3 の時間が全て一定となり、電流波形からのアタックを困難にする。あるいはタイマーには乱数発生器で生成した乱数をセットしておき、(c) に示すように毎回 t_1 , t_2 , t_3 の時間をランダムに変化させることも可能である。また、タイマーを用いなくても、ソフトウェアでカウントすることも可能である。

【 0 0 4 4 】

べき乗剰余乗算において、コプロセッサによる演算の高速化を目的とし、Y の値を 2 ビット、あるいは 3 ビットずつ処理するようにすると、例えば図 1 6 のフローチャート図に示すように、2 ビット処理の例で説明するなら常に $A^2 \bmod N - A^2 \bmod N - AB \bmod N$ 及び $i - 2$ と $i = 0 ?$ の各ステップの繰り返しになるので、前記 1 ビットずつ行なう場合のような攪乱目的のダミー演算を行なわなくとも、処理時間や電流波形が一定になる。そのため、電流波形から Y の値を推定するのは困難になる。また演算の回数も、前記のバイナリ法だと最大で $2n$ 回かかっていたものを、2 ビット処理だと常に $1.5n$ 回で済むために、動作時間の短縮にもつながる。

【 0 0 4 5 】

コプロセッサの演算が開始するまでに A , B , N の値をそれぞれコプロセッサ専用レジスタに転送し格納しておく。しかしながら、2 ビット処理を行う場合、Y の値によって 4 通りの B の値 B_1 , B_2 , B_3 , B_4 が必要になり、これら

の値は前もって計算して、RAMやEEPROMなどに格納しておき、毎回コプロセッサ専用レジスタに転送することになるが。この際、4通りのBの値によって転送中の電流波形に特徴が現れる可能性がある。

【0046】

例えば、16ビットのプリチャージバスにデータを転送する場合を考える。プリチャージバスは、データ転送の前にすべてのバスの値を“1”にそろえるバスである。このバスに、値は違うが“1”のビットの数が同じデータ、例えば、“1”のビットの数が2である16進数で“88”と“11”、を転送した場合、電流波形はほぼ同じ波形になると予測される。この理由は、“1”から“0”へ変化したビットの数が同じであるため、同じように電流を消費し、同じ電流波形になるからである。

【0047】

もし、“1”のビットの数が1つ異なるデータ、例えば、“1”のビットの数が3である“89”や“19”を転送した場合、“1”のビットの数が2のデータとは消費電流が異なる。これは、13ビット分バスの値が“1”から“0”に変わったため、その分の電流が消費される。そのため、先の14ビットが変化したデータに比べて消費電流が1ビット分小さくなる。一般に、変化するビットの数が多いほど電流波形は高くなるという規則性がある。この規則性から転送されているデータを推定することができると思われる、電流アタックの対象となりやすい。これを防ぐため次のような工夫を行なうものである。

【0048】

図17と図18には、この発明に係るコプロセッサの他の一実施例のブロック図がそれぞれ示されている。この実施例のコプロセッサは、2ビット処理と3ビット処理に向けられている。つまり、コプロセッサのレジスタ容量を増やして、2ビット処理の場合には4通りのBの値 $B_1 \sim B_4$ を、3ビット処理の場合には8通りのBの値 $B_1 \sim B_8$ をコプロセッサのレジスタに格納しておく。従って、演算の途中で記憶回路(RAM)からデータバスを通して上記コプロセッサのレジスタに前記のような転送の必要がなくなり、前記電流アタックに対して防御することができる。

【 0 0 4 9 】

制御用レジスタ (CCNT)

ビット 7	ビット 6		ビット 2	ビット 1	ビット 0
-	-		e_i	e_{i-1}

【 0 0 5 0 】

演算の種類

ビット 2	e_i	e_{i-1}	演算の種類
0	0	0	$A \leftarrow A^2 \bmod N$
0	1	0	$A \leftarrow A \bmod N$
0	1	1	$A \leftarrow A \times N$
1	0	0	$A \leftarrow AB_1 \bmod N$
1	0	1	$A \leftarrow AB_2 \bmod N$
1	1	0	$A \leftarrow AB_3 \bmod N$
1	1	1	$A \leftarrow AB_4 \bmod N$

【 0 0 5 1 】

つまり、前記図 1 6 に示したようなフローチャート図において、コプロセッサが $AB \bmod N$ を実行する際、下記のように 4 つ (3 ビット処理のときにはあるいは 8 つ) のうちの正しい B レジスタ CDB から値を選んで実行できるように、Y の 2 ビット (あるいは 3 ビット) の値をコプロセッサの制御レジスタ (CCNT) のビットに当てはめ、前記に示す制御レジスタ及び演算の種類のように、2 ビット処理の場合には、 $AB_1 \bmod N$ 、 $AB_2 \bmod N$ 、 $AB_3 \bmod N$ 、 $AB_4 \bmod N$ のうちどの演算をするかを選択させるようにする。

【 0 0 5 2 】

図 1 9 には、この発明に係るコプロセッサの他の一実施例のブロック図が示さ

れている。この実施例のコプロセッサも、2ビット処理や3ビット処理のような複数ビット処理に向けられている。この実施例では、データバスにスイッチを設けて演算をしながら転送できるようにする。この構成により、コプロセッサのレジスタ容量を増加させることなく、実行時間の短縮と電流アタック対策の両方に効果的である。

【 0 0 5 3 】

コプロセッサ専用レジスタ (CDA, CDB, CDN, CDW) は、同図に示すように4つのレジスタがCPUとコプロセッサの演算器との間で排他的に使用されている。2ビット処理を行う場合、2回の $A^2 \bmod N$ を行いながらその間にBの値をRAMからコプロセッサ専用レジスタユニット中のBレジスタCDBに転送できるようにすると効率的である。

【 0 0 5 4 】

コプロセッサのAレジスタCDAとBレジスタCDBのI/Oを分け、それぞれにリード/ライトバッファ (R/W Buffer) を設けて、それぞれ独立に動作できるようにする。演算器が $A^2 \bmod N$ を演算している間は、制御信号によりデータバスをパス1 (path 1) につなぎ、図示しないCPUのRAMからBの値を上記独立に設けられたリード/ライトバッファを介してBレジスタCDBに転送する。次に演算器が $AB \bmod N$ を実行する際には、制御信号によりパス2 (path 2) に切り換え、上記BレジスタのB値を演算器に送り上記CPUがBレジスタCDBにアクセスできないようにする。この方法を取ると、 $A^2 \bmod N$ を演算動作と、B値の転送動作が同時に行なわれるから演算時間が短縮されるだけでなく、演算と転送の消費電流が重なるため双方の波形が識別できなくなり、電流アタック対策に有効である。

【 0 0 5 5 】

図20には、この発明に係るICカード用チップの他の一実施例の要部ブロック図が示されている。この実施例では、暗号処理用演算ユニットとメモリ (RAM) 間の転送の際、メモリにカウンタを設けるようにするものである。この実施例では、2ビット処理に用いる4通りの値、あるいは3ビット処理に用いる8通りの値をコプロセッサ外部メモリRAMからコプロセッサ専用レジスタユニット

中の B レジスタ CDB に転送する際の電流攪乱を行なうようにするものである。

【 0 0 5 6 】

この実施例では、前記図 2 に示したような IC カード用チップにおいて、RAM の側にカウンタが設けられる。RAM は、カウンタで形成されたアドレス信号をデコードしてデータをデータバスに送出する。このとき、アドレスバスには、乱数発生器が形成された偽アドレスが送出される。これにより、アドレスとデータとの相関が無くなり、電流解析を困難とさせることができる。

【 0 0 5 7 】

図 2 1 には、上記カウンタの一実施例のブロック図が示されている。カウンタは、転送したいブロックの最初のアドレスを保持する先頭アドレスレジスタとインクリメンタを用い、ブロック転送をイネーブルにするイネーブル信号とクロック又はリード／ライト信号などによるインクリメント指示信号で制御する。ブロック転送を開始する際、まず転送の先頭アドレスと転送開始のイネーブル信号が CPU よりカウンタに送信され、上記先頭アドレスレジスタに保持される。その後は、インクリメント指示信号によって、インクリメンタが動作して先頭アドレスレジスタの先頭アドレス $A + 1$ を形成して、アドレスを生成するとともに上記先頭アドレスレジスタの内容を書き換えるので、図 2 2 のタイミング図に示すように、RAM アドレスが順番にインクリメント A 、 $A + 1$ 、 $A + 2$ 、 \dots されていき、そのアドレスに従って順次データ D_A 、 D_{A+1} 、 D_{A+2} \dots が書込まれ／読み出される。

【 0 0 5 8 】

この実施例では、ブロック転送がイネーブルになった後はアドレスバスからのアドレスをカウンタが受け付けなため、アドレスバスにどのような値が来ようとデータは正しく読み出されていく。従って、アドレスバスに乱数発生器などで生成した乱数 B 、 C 、 D 、 $E \dots$ が出力されるとアドレスバスの消費電流を攪乱でき、この効果からチップ全体の消費電流を攪乱できるため、チップ内部動作の解析を困難にすることが可能になる。

【 0 0 5 9 】

図 2 3 には、この発明に係る IC カード用チップの更に他の一実施例を示す要

部ブロック図が示されている。この実施例でも、暗号処理用演算ユニットとメモリ（RAM）間の転送の際、メモリにカウンタを設けるようにするものであが、かかる暗号処理用演算ユニットとメモリRAMの最初のアドレスをも攪乱するようアドレスオフセット機能が設けられる。つまり、乱数発生器などで生成した乱数をあらかじめCPUとカウンタ側に同時に転送しておき、ブロック転送の最初のアドレスに乱数を加えるか又は引くかした値をアドレスバスに出力する。カウンタ側ではアドレスバスの値を同じ乱数を用いて復号化し、最初のアドレスを得る。

【 0 0 6 0 】

図 2 4 には、上記転送動作を説明するためのタイミング図が示されている。乱数発生器で形成された乱数をあらかじめCPUとRAMに転送しておき、オフセット演算部 1 によりブロック転送の最初のアドレス A に乱数 S を加えるか引くかしたアドレス $A \pm S$ をアドレスバスに送出する。カウンタ側では、アドレスバスの値を同じ乱数 S を用いて復号化し、オフセット演算部 2 により最初のアドレス A を得て、以後前記同様にインクリメントしてアドレス $A + 1$ 、 $A + 2 \cdots$ を生成する。このようなアドレス $A + 1$ 、 $A + 2$ に同期して、乱数発生器が乱数 B、C、D \cdots をアドレスバスに送出するので、先頭のアドレスを含めてアドレスバスの消費電流を攪乱でき、チップ内部動作の解析をいっそう困難にすることが可能になる。

【 0 0 6 1 】

前記実施例のような暗号化／復号化装置において、べき乗剰余演算「 $X^Y \bmod N$ 」（X，Y，N は正の整数）を用いた場合、X，Y，N が、通常 1 0 0 ビット～2 0 0 0 ビット程度の非常に大きな数を使用されるため、「 $X^Y \bmod N$ 」をいかにして高速に実行するかが重要となる。その一つの解法として、剰余乗算「 $ABR^{-1} \bmod N$ 」を実行する次のようなアルゴリズムが知られており、本願出願人においては、特開平 1 0 - 2 1 0 5 7 号公報（米国登録番号 5, 9 6 1, 5 7 8）において、「 $ABR^{-1} \bmod N$ 」のアルゴリズムを基にした積和演算器を用いたマイクロコンピュータを提案している。

【 0 0 6 2 】

上記アルゴリズムは、次のステップ(1) ないし(12)からなる。

- (1) i n p u t $X, Y = e_n \ e_{n-1} \ \cdots \ e_1, N, R$
- (2) $B = R^2 \bmod N$
- (3) $A = X$
- (4) $A = A B R^{-1} \bmod N + k N$
- (5) $B = A$
- (6) f o r $i = n - 1$ t o 1 s t e p - 1 {
- (7) $A = A^2 R^{-1} \bmod N + k N$
- (8) i f $e_i = 1$ t h e n $A = A B R^{-1} \bmod N + k N$
- (9) }
- (10) $A = A R^{-1} \bmod N + k N$
- (11) $A = A \bmod N$
- (12) o u t p u t A

【 0 0 6 3 】

この発明の他の実施例では、前記図 2 コプロセッサ 2 0 9 において、上記アルゴリズム 5 のステップ(4) 、ステップ(7) 、ステップ(8) 、ステップ(10)で示された「 $A = A B R^{-1} \bmod N + k N$ 」等に記述される「剰余乗算」を実行するようにされる。かかるコプロセッサ 2 0 9 は、後述するような演算回路と制御回路が含まれる。剰余乗算の入力値 A, B, R, N 及び出力値 A は専用レジスタ又は RAM などの記憶装置に保持される。

【 0 0 6 4 】

図 2 6 には、この発明に用いられるコプロセッサの他の一実施例のブロック図が示されている。同図において 3 3 は第 1 の積和演算器、3 4 は第 2 の積和演算器、3 5 は一次記憶値 $T e m p$ を保持するテンポラリレジスタ、3 6 は値 A の格納に利用されるレジスタ、3 7 は値 B の格納に利用されるレジスタ、3 8 は値 N の格納に利用されるレジスタである。3 9 は M_i 生成ロジック、4 0 は M_i 生成ロジック 3 9 で生成された値 M_i を保持するラッチ、4 1 は「 $\div 2^L$ 」を行うためのシフト回路である。

【 0 0 6 5 】

この実施例では、前記公報に詳細に説明されているようなブロック分割に基づいて演算「 $(AB_i + M_i N)/2^L$ 」を実行するようにされる。先ず、第1の積和演算器33は、レジスタ35の値Temp、レジスタ36の値A、レジスタ37の値 B_i を入力として、積和演算「 $Temp + A \cdot B_i$ 」を実行する。その演算結果は値Temp2として次段の第2の積和演算器34へ送られる。上記値Temp2は $n+L$ ビット長の整数である。

【0066】

一方、Mi生成ロジック39は、Lビット長の数 A_0 、 B_i 、 N_0 を入力としてLビットの整数 M_i を生成し、この正数 M_i はレジスタ40に一時的に保持される。第2の積和演算器34は、前記Temp2、N、 M_i を入力として、積和演算「 $Temp2 + M_i \cdot N$ 」を実行する。 $n+L$ ビット長の演算結果の下位Lビットは全て0であり、これをシフタ41によって消去して（すなわち 2^L で割って）、nビット長の結果が値Tempとしてレジスタ35に送られて保持される。

【0067】

以上の動作を n/L 回繰り返し実行すれば、演算「 $(AB+MN)/R$ 」が実現できる。これによれば、nビットの整数Mをあらかじめ計算して保持する必要はなく、Lビット長の M_i のみを積和演算器33の計算中に求めてレジスタ40に保持すればよく、値Mの計算時間の削除、および値Mを保持する記憶手段の規模を縮小することができる。さらに、積和演算器33と積和演算器34を直列的に接続して連続的に動作させることにより、 $n+L$ ビット長の中間結果Temp2を一時的に保持する記憶手段を特別に設けることも必要なくなる。

【0068】

レジスタ35～38を積和演算器33、34にバス43で接続される。したがって前記レジスタ35～38をRAM42で構成することができる。これにより、半導体チップ上のレジスタ面積の低減が可能となる。また、この構成においては、特にデータバス43によるデータ転送量が多いため、バス幅が大きくなって半導体チップの面積が大きくなるようにする必要が生じるが、図26の実施例のように積和演算器33と積和演算器34を直列的に接続することにより、中

間結果Temp 2をデータバスを用いて転送することが不要になるため、バスによるデータ転送量の低減を図ることができる。

【0069】

この実施例のコプロセッサでは、第1の積和演算器33でTemp = 0、第2の積和演算器34で $M_i \cdot N = 0$ 、さらにセレクタ41による「 $\div 2^L$ 」の動作を行なわないことにより、同図に示される演算手段を、「 $A \cdot B_i$ 」のような多倍長乗算（小さな数 B_i とその多倍長に相当する大きな数Aとの乗算）を実行する回路として使用することができる。「 $A \cdot B_i$ 」のような多倍長乗算演算は、上記アルゴリズムのステップ(2)の演算「 $R^2 \bmod N$ 」をマイクロプロセッサ201を用いて実行するときに適用されることにより、その演算の高速化を図ることができる。

【0070】

図27の「 $R^2 \bmod N$ 」の計算の概念図に示されているように、 $R = 2^n$ 、 $n = 512$ とされ、 N は512ビット、 R^2 は最上位ビットだけが1で下位側1024ビット全てが0の値とされる。マイクロプロセッサで演算「 $R^2 \bmod N$ 」を行うとき、大きな数の R^2 を同様に大きな数の N で直接に除算するのは効率的でないから、被除数を最上位側から64ビット単位のブロックとして把握し、また、除数を最上位側から32ビット単位のブロックとして把握し、順次上位側のブロック同士を対象に除算を行い、それによって得られる値を商の概数として把握する。

【0071】

同図において、例えば $Q (= Da \div Na)$ を商の概数として把握する。概略的には、 R^2 の上位側に対して「 $Q \cdot Na$ 」を減算し、その減算結果の上位側に対して「 $Q \cdot Nb$ 」を減算する。「 $Q \cdot Nb$ 」の減算結果に対して同様の処理を行い、更にその結果の対して同様の処理を繰り返すという手法によって、「 $R^2 \bmod N$ 」の結果を得ることができる。

【0072】

実際にはその途上で、余剰ビットを消去するための減算処理が介在される。このとき、前記演算「 $Q \cdot Nb$ 」の処理は、第1回目では32ビットと480ビッ

トという大きな数の乗算処理とされる。しかもそのような大きな数の乗算処理は何回も繰り返される。このとき、前記図 2 6 に示されるコプロセッサによって演算可能な前記「 $A \cdot B_i$ 」のような多倍長乗算演算を利用することにより、換言すれば、そのような多倍長乗算演算をコプロセッサに負担させれば、上記アルゴリズム 5 におけるステップ 2 の演算「 $R^2 \bmod N$ 」をマイクロプロセッサ 2 0 1 を用いて実行するとき、その演算処理の高速化を図ることができる。

【 0 0 7 3 】

前記のようなアルゴリズムにおける「 $A = A B R^{-1} \bmod N$ 」の演算処理では、前記公報（特開平 1 0 - 2 1 0 5 7 号）において詳述されているように、剰余乗算において、オーバーフロー有りのときには更に演算結果 W から N を減算 $W - N$ するものであるため、オーバーフローの有無により演算時間や消費電流の違いが生じる。このため、前記のような IC カード L S I の消費電流を観測し、そのタイミングや統計的処理の結果からチップ内の動作を解析されてしまう可能性を持っている。

【 0 0 7 4 】

図 2 8 には、この発明に係る暗号化処理用演算ユニットの一実施例の要部ブロック図が示されている。この実施例の暗号化処理用演算ユニットは、前記のような IC カード等に搭載される 1 チップのマイクロコンピュータに含まれるコプロセッサに含まれる。

【 0 0 7 5 】

図 2 8 において、前記図 2 6 に示した第 1 と第 2 の積和演算器 3 3、3 4 を含む積和演算器により、前記 $A^2 R^{-1} \bmod N$ 又は $A B R^{-1} \bmod N$ の演算が行われ、その演算結果 W はテンポラリレジスタ $T e m p$ に格納され、演算結果にオーバーフローが発生した場合には演算器からのオーバーフローフラグ $O V$ が制御論理の $O V$ 格納レジスタに記憶される。そして、続いて上記剰余乗算の後にテンポラリレジスタ $T e m p$ に格納された演算結果 $W - N$ の減算が行われる。

【 0 0 7 6 】

上記オーバーフローフラグ $O V$ が有る時（論理 1）には上記減算 $W - N$ の結果は、テンポラリレジスタ $T e m p$ に格納され、オーバーフローフラグ $O V$ が無い

時（論理0）には減算 $W-N$ の結果は、テンポラリレジスタ $Temp$ に格納されず、上記テンポラリレジスタ $Temp$ 以外の適当な記憶回路、例えばレジスタ A に格納される。つまり、前記減算 $W-N$ と、それにより形成された無効データを適当な記憶回路に格納する動作は、前記錯乱目的のダミー動作とされる。これにより、前記剰余乗算においてオーバーフローが生じないときでも、 $W-N$ の減算及びその演算結果をレジスタに格納することに伴うICカードの動作電流が常に発生し、オーバーフローの有無を外部より識別困難とすることができるものとなる。

【0077】

上記の信号処理は、次のようなプログラムによって実施される。

```

W ← (AB+MN) / R
Store OV bit
if OV then
    W ← W-N    (正規のオーバーフロー処理とWへの書き込み)
Else
    A ← W-N    (ダミーのオーバーフロー処理とAへの書き込み)
Exchange W and A
Output A

```

【0078】

上記プログラムにおいて、 W はテンポラリレジスタ及びそのデータを表している。そして、オーバーフローフラグ OV 無しの際に、テンポラリレジスタ $Temp$ のアドレスをレジスタ A のアドレスを交換することにより、オーバーフローフラグ OV の有リ／無しに対応して W 又は A のデータが有効なデータとして出力される。この実施例では、 $Exchange\ W\ and\ A$ のようなアドレス交換によって、レジスタ A のアドレス指定によりテンポラリレジスタ（ W ）のデータを出力させるものである。

【0079】

上記の信号処理は、次のようなプログラムに置き換えることができる。

```

W ← (AB+NM) / R

```

Store OV bit

$A \leftarrow W - N$ (オーバーフロー処理とレジスタAへの書き込み)

if! OV then

Exchange W and A

Else nop

Output A

【0080】

つまり、オーバーフローOVの有無に無条件でのオーバーフロー処理のための減算 $W - N$ とその減算結果をAレジスタの書き込みを行った後に、オーバーフローフラグOVが無ければ、Exchange W and Aのようにアドレス交換を行ってレジスタAのアドレス指定によりテンポラリレジスタ(W)のデータを出力させ、無ければアドレスを交換することなくAレジスタのデータ $W - N$ を有効なデータとして出力させる。

【0081】

上記の信号処理は、更に次のようなプログラムに置き換えることができる。

$W \leftarrow (AB + NM) / R$

Store OV bit

Exchange W and A

$W \leftarrow A - N$ (オーバーフロー処理とレジスタAへの書き込み)

if OV then

Exchange W and A

Else nop

Output A

【0082】

つまり、オーバーフローOVの有無に無条件でのオーバーフロー処理のための減算 $W - A$ を行う前にExchange W and Aのようにアドレス交換を行って $A - N$ の減算、つまりは $W - N$ の減算を行ってテンポラリレジスタ(W)、つまりレジスタAにデータを出力させる。そして、オーバーフローフラグOVが有れば、Exchange W and Aのように再度アドレス交換を行

い、レジスタ A のアドレス指定によりレジスタ A のデータを、オーバーフローフラグ O V が無ければ前記のように交換したままレジスタ A のアドレス指定によりテンポラリレジスタ (W) のデータを出力させる。この構成では、レジスタへの書き込みを行う論理回路は、みかけ上テンポラリレジスタ (W) を書き込むような構成となり、レジスタ A への書き込み用論理が不要となって回路の簡素化が可能になる。

【 0 0 8 3 】

前記テンポラリレジスタ T e m p とレジスタ A のアドレス交換は、フラグ反転回路により実現できる。つまり、アドレスバスから供給されるアドレス信号のうち、例えば最下位ビットのような 1 ビットがテンポラリレジスタ T e m p とレジスタ A とで異なるように設定しておき、フラグ反転回路により選択的にかかるビットを交換するだけで、テンポラリレジスタ T e m p に割り当てられたアドレス指定によりレジスタ A を選択でき、逆にレジスタ A に割り当てられたアドレスによりテンポラリレジスタ T e m p を選択することができる。

【 0 0 8 4 】

図 2 8 の実施例において、2 つのレジスタ T e m p と A を使い、オーバーフローフラグ O V に対応して、常に一方（例えばテンポラリレジスタ T e m p ）に有効データを格納させ、前記のようなアドレス交換によってレジスタ A を指定するアドレスにより有効なデータを出力させる。前記減算 W - N と、それにより形成された無効データを適当な記憶回路に格納する動作が錯乱目的のダミー動作とされる。これにより、前記剰余乗算においてオーバーフローが生じないときでも、W - N の減算及びその演算結果をレジスタに格納することに伴う I C カードの動作電流が常に発生し、オーバーフローの有無を外部より識別困難とすることができるものとなる。

【 0 0 8 5 】

前記実施例のように積和演算器の前記のようなオーバーフローフラグに代えてボロー (B o r r o w) フラグ B R を利用するものであってもよい。つまり、前記 $A^2 R_j^{-1} \bmod N$ 又は $A B R^{-1} \bmod N$ の演算結果 W はテンポラリレジスタ T e m p に格納し、W - N の減算が行われときの演算器からのボローフラグ B R を

記憶し、ボローフラグBRが有るときのみ、テンポラリレジスタTempとレジスタAのアドレスを交換し、最終的にはレジスタAのアドレス指定により有効なデータを読み出すようにするものであってもよい。

【0086】

上記の信号処理は、次のようなプログラムにより実現できる。

```
W ← (AB+NM) / R
A ← W - N
Store BR bit
if BR then
    Exchange W and A
Else nop
Output A
```

【0087】

図29には、この発明に係る暗号化処理用演算ユニットの他の一実施例の要部ブロック図が示されている。この実施例の暗号化処理用演算ユニットも前記のようなICカード等に搭載される1チップのマイクロコンピュータに含まれるコプロセッサに含まれる。この実施例では、データバスの信号と前記のような積和演算器の出力とのうちのいずれか一方を前記のようなOVフラグ格納レジスタに記憶されたオーバーフローフラグOVに従って出力させるセクタが追加される。

【0088】

上記剰余乗算の後W-Nの減算が行われ、この減算結果W-Nと演算のためにデータバス上に読み出されたWの値がセクタに入力され、オーバーフローフラグOVが有るときには減算結果W-Nが、オーバーフローフラグOVが無いときにはデータバスWの値が選択され、この選択された値がレジスタAに格納され、Aが最終的に有効なデータとして出力されることにより、W-Nの減算とレジスタへの書込みに伴うICカードやマイクロコンピュータの動作電流が常に発生し、オーバーフローの有無を外部より識別困難とされる。

【0089】

図30には、この発明に係る暗号化処理用演算ユニットの更に他の一実施例の

要部ブロック図が示されている。この実施例は、前記図 2 8 の実施例においてレジスタブロックにレジスタ X が追加される。前記同様に剰余乗算の後 $W - N$ の減算が行われ、オーバーフローフラグ OV が有る時にはこの減算結果 $W - N$ がレジスタ A に、オーバーフローフラグ OV が無い時にはダミー演算専用のレジスタ X に減算 $W - N$ が書込まれる。この後に、オーバーフローフラグ OV が無い時には、テンポラリレジスタ (W) とレジスタ A とのアドレスを交換して、最終的にレジスタ A を選択するアドレスにより有効なデータを出力させる。

【0090】

上記の信号処理は、次のようなプログラムにより実現できる。

```

W ← (AB + NM) / R
Store OV bit
if OV then
    A ← W - N    (正規のオーバーフロー処理と A への書き込み)
Else
    X ← W - N    (ダミーのオーバーフロー処理と X への書き込み)
Exchange W and A
Output A

```

【0091】

上記の実施例から得られる作用効果は、下記の通りである。すなわち、

(1) 外部端子がリードライト装置と電氣的に接続されることによって動作電圧が供給され、かつ、暗号化処理又は復号化処理を伴ったデータの入出力動作を含む IC カードにおいて、上記暗号化処理又は復号化処理に本来の処理動作に似た攪乱目的のダミー処理動作を含ませて内部回路の動作タイミング及び動作電流の画一化を行なうようにすることによって、電流波形を利用した解読を困難にすることができるという効果が得られる。

【0092】

(2) 上記に加えて、上記暗号化処理又は復号化処理は、RSA 暗号法などに応用可能なべき乗剰余乗算動作を含むようにすることにより、機密保護の強化を実現した IC カードを得ることができるという効果が得られる。

【 0 0 9 3 】

(3) 上記に加えて、上記べき乗剰余演算動作を中央処理装置からの指示を受けて動作する暗号処理用演算ユニットにより行わせることにより、高速なデータ処理を行なうようにすることができるという効果が得られる。

【 0 0 9 4 】

(4) 上記に加えて、上記暗号化処理用演算ユニットの動作として、入力された X 、 Y 及び N を受け、 $A = 1$ 、 $B = X$ として、 $A = A^2 \bmod N$ と $A = AB \bmod N$ の演算を交互行ない、かかる演算において Y の上位から 1 ビットずつみて、論理 0 であれば上記 $A^2 \bmod N$ の演算結果を有効なデータとして記憶回路に取り込み、論理 1 であれば上記 $A^2 \bmod N$ と $AB \bmod N$ の演算結果を有効なデータとして記憶回路に取り込むものとし、上記論理 0 のときの $A = AB \bmod N$ の演算動作を上記攪乱目的のダミー処理動作とすることにより、暗号処理を行いつつ電流波形を利用した解読を困難にすることができるという効果が得られる。

【 0 0 9 5 】

(5) 上記に加えて、上記記憶回路をリードライトバッファを通してデータの入出力が行なわれる複数のレジスタからなるレジスタブロックを用い、上記 Y の特定ビット e_i の論理 1 又は 0 によってゲート回路を制御し、所定のレジスタに供給されるライトストローク信号の伝達を制御して、上記演算結果のうち有効なデータのみがリードライトバッファを通して上記所定のレジスタに格納することにより、暗号処理を行いつつ電流波形を利用した解読を困難にすることができるという効果が得られる。

【 0 0 9 6 】

(6) 上記に加えて、上記記憶回路をリードライトバッファを通してデータの入出力が行なわれる複数のレジスタとからなるレジスタブロックを用い、上記 Y の特定ビット e_i の論理 1 又は 0 によってゲート回路を制御し、上記リードライトバッファに供給されるライトストローク信号の伝達を制御して、上記演算結果のうち有効なデータのみがリードライトバッファを通して上記所定のレジスタに格納することにより、暗号処理を行いつつ電流波形を利用した解読を困難にする

ことができるという効果が得られる。

【 0 0 9 7 】

(7) 上記に加えて、上記記憶回路をリードライトバッファを通してデータの入出力が行なわれる複数のレジスタ及びダミーレジスタとからなるレジスタブロックを用い、上記リードライトバッファと上記ダミーレジスタ及び複数のレジスタとの間セレクトを設けて上記Yの特定ビット e_i の論理1又は0によって制御して、リードライトバッファに書き込まれた演算結果のうち有効なデータを所定のレジスタに格納し、無効なデータが上記ダミーレジスタに格納することにより、暗号処理を行いつつ電流波形を利用した解読をよりいっそう困難にすることができるという効果が得られる。

【 0 0 9 8 】

(8) 上記に加えて、上記暗号化処理用演算ユニットの動作として、入力されたX、Y及びNを受け、 $A=1$ 、 $B=X$ として、 $A=A^2 \bmod N$ と $A=AB \bmod N$ の演算を交互行ない、かかる演算においてYの上位から1ビットずつみて、論理0であれば上記 $A^2 \bmod N$ の演算結果を有効なデータとしてその出力タイミングで記憶回路に取り込み、論理1であれば上記 $A^2 \bmod N$ と $AB \bmod N$ の演算結果を有効なデータとしてその出力タイミングで記憶回路に取り込むものであり、上記 $A=A^2 \bmod N$ の演算結果が出力されてから上記 $A=AB \bmod N$ の演算が開始されるまでの間も上記 $A=A^2 \bmod N$ の動作を継続し、 $A=AB \bmod N$ の演算結果が出力されてからYのビットの変更判定処理を含めて次のビットに対応した $A^2 \bmod N$ の演算が開始されるまでの間も上記 $A=AB \bmod N$ の動作を継続することにより、暗号処理を行いつつ電流波形を利用した解読をよりいっそう困難にすることができるという効果が得られる。

【 0 0 9 9 】

(9) 上記に加えて、上記暗号化処理用演算ユニットの動作として、入力されたX、Y及びNを受け、 $A=1$ 、 $B=X$ として、 $A=A^2 \bmod N$ と $A=AB \bmod N$ の演算とそれぞれに対してオーバーフロー演算行ない、かかる演算においてYの上位から1ビットずつみて、論理0であれば上記 $A^2 \bmod N$ の演算結果を有効なデータとして記憶回路に取り込み、論理1であれば上記 $A^2 \bmod N$ と

A B m o d N の演算結果を有効なデータとして記憶回路に取り込むものであり、上記論理 0 のときの $A = A B m o d N$ の演算動作と、各演算動作での不要なオーバーフロー演算を上記攪乱目的のダミー処理動作とすることにより、暗号処理を行いつつ電流波形を利用した解読をよりいっそう困難にすることができるという効果が得られる。

【0100】

(10) 外部端子がリードライト装置と電氣的に接続されることによって動作電圧が供給され、かつ、暗号化処理又は復号化処理を伴ってデータの入出力動作が行われる IC カードに、上記暗号化処理又は復号化処理に攪乱目的のダミー演算を含ませて内部回路の動作タイミング及び動作電流に不規則性を持たせることにより、暗号処理を行いつつ電流波形を利用した解読をよりいっそう困難にした IC カードを得ることができるという効果が得られる。

【0101】

(11) 外部端子がリードライト装置と電氣的に接続されることによって動作電圧が供給され、かつ、暗号化処理又は復号化処理を伴ってデータの入出力動作が行われる IC カードに、上記暗号化処理又は復号化処理における各演算の間隔に攪乱目的のダミーサイクルを含ませて内部回路の動作タイミング及び動作電流に不規則性を持たせることにより、暗号処理を行いつつ電流波形を利用した解読をよりいっそう困難にした IC カードを得ることができるという効果が得られる。

【0102】

(12) 暗号化処理又は復号化処理を伴ったデータの入出力動作を含むモジュール構成のマイクロコンピュータにおいて、上記暗号化処理又は復号化処理に攪乱目的のダミー処理動作を含ませて内部回路の動作タイミング及び動作電流の画一化を行なうようにすることにより、モジュール化されたマイクロコンピュータに対する電流波形を利用した解読を困難にすることができるという効果が得られる。

【0103】

(13) 上記に加えて、上記マイクロコンピュータのモジュール構成を 1 つの

半導体基板上において形成することにより、小型化を図りつつ電流波形以外の直接的なプログラム又はデータ等の解読も防止することができるという効果が得られる。

【0104】

(14) 上記に加えて、上記マイクロコンピュータの暗号化処理又は復号化処理を、RSA暗号法などに応用可能なべき乗剰乗算動作を含むものとし、上記べき乗剰乗算動作を中央処理装置からの指示を受けて動作する暗号処理用演算ユニットにより行なうようにすることにより、高速な暗号処理動作を行なうようにすることができるという効果が得られる。

【0105】

(15) 上記に加えて、上記マイクロコンピュータの暗号化処理用演算ユニットの動作として、入力されたX、Y及びNを受け、 $A=1$ 、 $B=X$ として、 $A=A^2 \bmod N$ と $A=AB \bmod N$ の演算を行ない、かかる演算においてYの上位から1ビットずつみて、論理0であれば上記 $A^2 \bmod N$ の演算結果を有効なデータとして記憶回路に取り込み、論理1であれば上記 $A^2 \bmod N$ と $AB \bmod N$ の演算結果を有効なデータとして記憶回路に取り込むものとし、上記論理0のときの $A=AB \bmod N$ の演算動作を上記攪乱目的のダミー処理動作とすることにより、暗号処理を行いつつ電流波形を利用した解読を困難にすることができるという効果が得られる。

【0106】

(16) 上記に加えて、上記マイクロコンピュータの暗号化処理用演算ユニットの動作として、入力されたX、Y及びNを受け、 $A=1$ 、 $B=X$ として、 $A=A^2 \bmod N$ と $A=AB \bmod N$ の演算を行ない、かかる演算においてYの上位から1ビットずつみて、論理0であれば上記 $A^2 \bmod N$ の演算結果を有効なデータとしてその出力タイミングで記憶回路に取り込み、論理1であれば上記 $A^2 \bmod N$ と $AB \bmod N$ の演算結果を有効なデータとしてその出力タイミングで記憶回路に取り込むものであり、上記 $A=A^2 \bmod N$ の演算結果が出力されてから上記 $A=AB \bmod N$ の演算が開始されるまでの間も上記 $A=A^2 \bmod N$ の動作を継続し、 $A=AB \bmod N$ の演算結果が出力されてからYのビットの

変更判定処理を含めて次のビットに対応した $A^2 \bmod N$ の演算が開始されるまでの間も上記 $A = AB \bmod N$ の動作を継続することにより、暗号処理を行いつつ電流波形を利用した解読を困難にすることができるという効果が得られる。

(17) 上記に加えて、上記マイクロコンピュータの暗号化処理用演算ユニットの動作として、入力された X 、 Y 及び N を受け、 $A = 1$ 、 $B = X$ として、 $A = X^Y \bmod N$ と $A = AB \bmod N$ の演算とそれぞれに対してオーバーフロー演算を行ない、かかる演算において Y の上位から1ビットずつみて、論理0であれば上記 $A^2 \bmod N$ の演算結果を有効なデータとして記憶回路に取り込み、論理1であれば上記 $A^2 \bmod N$ と $AB \bmod N$ の演算結果を有効なデータとして記憶回路に取り込むものであり、上記論理0のときの $A = AB \bmod N$ の演算動作と、各演算動作での不要なオーバーフロー演算を上記攪乱目的のダミー処理動作とすることにより、暗号処理を行いつつ電流波形を利用した解読を困難にすることができるという効果が得られる。

【0107】

(18) 上記に加えて、上記暗号化処理用演算ユニットにより、入力された X 、 Y 及び N を受け、 $A = 1$ 、 $B = X$ として、 Y のビットの値に応じて、 $A = A^2 R^{-1} \bmod N$ 、 $A = AB R^{-1} \bmod N$ の演算行うとともに、演算結果にオーバーフローが発生した場合にはさらに上記演算結果 W から N の減算 $W - N$ を行なう正規動作と、各々の演算結果にオーバーフローが発生しない場合でも上記減算 $W - N$ に対応した演算による無効データを生成する攪乱目的のダミー動作を行い、上記オーバーフローの有無に対応して有効なデータを出力させることにより、暗号化処理用演算ユニットの簡素化及び高速化を図りつつ、電流波形を利用した解読を困難にすることができるという効果が得られる。

【0108】

(19) 上記に加えて、上記 $A^2 R^{-1} \bmod N$ 又は $AB R^{-1} \bmod N$ の演算結果 W を第1の記憶回路に格納し、演算器のオーバーフローフラグ OV の有無を記憶し、上記剰余乗算の後に上記第1記憶回路の演算結果 W から N の減算 $W - N$ を行い、その演算結果を上記オーバーフローフラグ OV が有る時には上記第1の記憶回路に格納し、オーバーフローフラグ OV が無い時には上記第1記憶回路とは

異なる第 2 の記憶回路に上記錯乱目的のダミー動作として格納し、上記第 1 の記憶回路の演算結果を有効なデータとして出力させることにより、暗号化処理用演算ユニットの簡素化及び高速化を図りつつ、電流波形を利用した解読を困難にすることができるという効果が得られる。

【 0 1 0 9 】

(2 0) 上記に加えて、上記 $A^2 R^{-1} \bmod N$ 又は $ABR^{-1} \bmod N$ の演算結果 W を第 1 の記憶回路に格納し、演算器のオーバーフローフラグ OV の有無を記憶し、上記剰余乗算の後に上記第 1 の記憶回路の演算結果 W から N の減算 $W - N$ を行い、オーバーフローフラグ OV が有るときに上記演算結果 $W - N$ をセレクタにより選択され、オーバーフローフラグ OV が無いときには上記第 1 記憶回路の演算結果 W をセレクタにより選択して第 2 の記憶回路に格納することにより、暗号化処理用演算ユニットの簡素化及び高速化を図りつつ、電流波形を利用した解読を困難にすることができるという効果が得られる。

【 0 1 1 0 】

(2 1) 上記に加えて、上記 $A^2 R^{-1} \bmod N$ 又は $ABR^{-1} \bmod N$ の演算結果 W を第 1 の記憶回路に格納し、演算器のオーバーフローフラグ OV の有無を記憶し、上記剰余乗算の後に上記第 1 の記憶回路の演算結果 W から N の減算 $W - N$ を行い、オーバーフローフラグ OV が有るときには減算 $W - N$ を第 2 の記憶回路に記憶し、オーバーフローフラグ OV が無いときには減算 $W - N$ を第 3 の記憶回路に記憶し、オーバーフローフラグ OV が有るときには上記第 2 の記憶回路のデータが有効なデータとして出力し、オーバーフローフラグ OV が無いときには上記第 1 の記憶回路のデータが有効なデータとして出力することにより、暗号化処理用演算ユニットの簡素化及び高速化を図りつつ、電流波形を利用した解読を困難にすることができるという効果が得られる。

【 0 1 1 1 】

(2 2) 上記に加えて、上記 $A^2 R^{-1} \bmod N$ 又は $ABR^{-1} \bmod N$ の演算結果 W を第 1 の記憶回路に格納し、演算器のオーバーフローフラグ OV の有無を記憶し、上記剰余乗算の後に上記第 1 の記憶回路の演算結果 W から N の減算結果 $W - N$ を第 2 の記憶回路に格納し、オーバーフローフラグ OV が無いとき第 1 の記

憶回路と第 2 の記憶回路を選択する最下位アドレスを逆転させて、上記第 2 の記憶回路を選択するアドレスにより第 1 の記憶回路を選択して有効なデータとして出力させ、オーバーフローフラグ OV が有るとき第 1 の記憶回路と第 2 の記憶回路を選択する最下位アドレスをそのままにして第 2 の記憶回路の演算結果を有効なデータとして出力させることにより、暗号化処理用演算ユニットの簡素化に加えてレジスタへの書き込み論理の簡素化と高速化を図りつつ電流波形を利用した解読を困難にすることができるという効果が得られる。

【 0 1 1 2 】

(2 3) 上記に加えて、上記 $A^2 R^{-1} \bmod N$ 又は $ABR^{-1} \bmod N$ の演算結果 W を第 1 の記憶回路に格納し、演算器のオーバーフローフラグ OV の有無を記憶し、上記剰余乗算の後に上記第 1 の記憶回路と第 2 の記憶回路のアドレスを交換し、第 2 の記憶回路を選択するアドレスにより選択される演算結果値 W から N の減算 $W - N$ が行われて第 1 の記憶回路を選択するアドレスにより選択される第 2 の記憶回路に減算結果 $W - N$ を格納し、オーバーフローフラグ OV が有るときにのみ上記アドレスを再度交換し、第 1 の記憶回路を選択するアドレスにより選択される第 1 又は第 2 の記憶回路のデータを有効なデータとして出力させることにより、暗号化処理用演算ユニットの簡素化に加えてレジスタへの書き込み論理の簡素化と高速化を図りつつ電流波形を利用した解読を困難にすることができるという効果が得られる。

【 0 1 1 3 】

(2 4) 上記に加えて、上記 $A^2 R^{-1} \bmod N$ 又は $ABR^{-1} \bmod N$ の演算結果 W を第 1 の記憶回路に格納し、上記剰余乗算の後に上記第 1 の記憶回路の演算結果値 W から N の減算 $W - N$ を行って第 2 の記憶回路に格納し、この $W - N$ の減算が行われた時の演算器のボロフフラグ BR を記憶し、ボロフフラグ BR が有るときには、第 1 の記憶回路と第 2 の記憶回路を選択する最下位アドレスを逆転させて、上記第 2 の記憶回路を選択するアドレスにより第 1 の記憶回路の演算結果 W を出力し、ボロフフラグ BR が無いときには、第 1 の記憶回路と第 2 の記憶回路を選択する最下位アドレスをそのままにして、上記第 2 の記憶回路を選択するアドレスにより第 2 の記憶回路の演算結果 $W - N$ を出力させることにより、暗号化

処理用演算ユニットの簡素化に加えてレジスタへの書き込み論理の簡素化と高速化を図りつつ電流波形を利用した解読を困難にすることができるという効果が得られる。

【 0 1 1 4 】

(25) 上記に加えて、上記マイクロコンピュータの暗号化処理用演算ユニットとして、入力されたX、Y及びNを受け、 $A = 1$ 、 $B = X$ として、Yのビットの値に応じて、 $A = A^2 R^{-1} \bmod N$ 、 $A = A B R^{-1} \bmod N$ の演算を行い、演算結果にオーバーフローが発生した場合にはさらに上記演算結果WからNの減算 $W - N$ を行なう正規動作と、各々の演算結果にオーバーフローが発生しない場合でも上記減算 $W - N$ に対応した演算による無効データを生成する攪乱目的のダミー動作を行い、上記オーバーフローの有無に対応して有効なデータを出力させることにより、暗号化処理用演算ユニットの簡素化に加えてレジスタへの書き込み論理の簡素化と高速化を図りつつ電流波形を利用した解読を困難にすることができるという効果が得られる。

【 0 1 1 5 】

(26) 上記に加えて、上記マイクロコンピュータの暗号化処理用演算ユニットにおいて、上記 $A^2 R^{-1} \bmod N$ 又は $A B R^{-1} \bmod N$ の演算結果Wは第1の記憶回路に格納し、演算器からのオーバーフローフラグOVの有無を記憶し、上記剰余乗算の後に上記第1記憶回路の演算結果WからNの減算 $W - N$ を行ない、演算結果を上記オーバーフローフラグOVが有る時には上記第1の記憶回路に格納し、オーバーフローフラグOVが無い時には上記第1記憶回路とは異なる第2の記憶回路に上記錯乱目的のダミー動作として書き込み、上記第1の記憶回路の演算結果が有効なデータとして出力させることにより、暗号化処理用演算ユニットの簡素化及び高速化を図りつつ、電流波形を利用した解読を困難にすることができるという効果が得られる。

【 0 1 1 6 】

(27) 上記に加えて、上記マイクロコンピュータの暗号化処理用演算ユニットにおいて、上記 $A^2 R^{-1} \bmod N$ 又は $A B R^{-1} \bmod N$ の演算結果Wを第1の記憶回路に格納し、演算器のオーバーフローフラグOVの有無を記憶し、上記剰

余乗算の後に上記第 1 の記憶回路の演算結果 W から N の減算 $W - N$ を行ってオーバーフローフラグ OV が有るときにセレクトにより上記演算結果 $W - N$ を選択し、オーバーフローフラグ OV が無いときにはセレクトにより上記第 1 記憶回路の演算結果 W を選択して第 2 の記憶回路に格納して有効なデータとして出力することにより、暗号化処理用演算ユニットの簡素化及び高速化を図りつつ、電流波形を利用した解読を困難にすることができるという効果が得られる。

【 0 1 1 7 】

(28) 上記に加えて、上記マイクロコンピュータの暗号化処理用演算ユニットにおいて、上記 $A^2 R^{-1} \bmod N$ 又は $ABR^{-1} \bmod N$ の演算結果 W を第 1 の記憶回路に格納し、演算器のオーバーフローフラグ OV の有無を記憶し、上記剰余乗算の後に上記第 1 の記憶回路の演算結果 W から N の減算 $W - N$ を行ってオーバーフローフラグ OV が有るときには減算結果 $W - N$ を第 2 の記憶回路に記憶し、オーバーフローフラグ OV が無いときには減算結果 $W - N$ を第 3 の記憶回路に記憶し、オーバーフローフラグ OV が有るときには上記第 2 の記憶回路のデータが有効なデータとして出力し、オーバーフローフラグ OV が無いときには上記第 1 の記憶回路のデータを有効なデータとして出力することにより、暗号化処理用演算ユニットの簡素化及び高速化を図りつつ、電流波形を利用した解読を困難にすることができるという効果が得られる。

【 0 1 1 8 】

(29) 上記に加えて、上記マイクロコンピュータの暗号化処理用演算ユニットにおいて、上記 $A^2 R^{-1} \bmod N$ 又は $ABR^{-1} \bmod N$ の演算結果 W を第 1 の記憶回路に格納し、演算器のオーバーフローフラグ OV の有無を記憶し、上記剰余乗算の後に上記第 1 の記憶回路の演算結果 W から N の減算結果 $W - N$ を第 2 の記憶回路に格納し、オーバーフローフラグ OV が無いとき第 1 の記憶回路と第 2 の記憶回路を選択する最下位アドレスを逆転させて、上記第 2 の記憶回路を選択するアドレスにより第 1 の記憶回路を選択して有効なデータとして出力させ、オーバーフローフラグ OV が有るとき第 1 の記憶回路と第 2 の記憶回路を選択する最下位アドレスをそのままにして第 2 の記憶回路の演算結果を有効なデータとして出力させることにより、暗号化処理用演算ユニットの簡素化に加えてレジスタ

への書き込み論理の簡素化と高速化を図りつつ電流波形を利用した解読を困難にすることができるという効果が得られる。

【 0 1 1 9 】

(30) 上記に加えて、上記マイクロコンピュータの暗号化処理用演算ユニットにおいて、上記 $A^2 R^{-1} \bmod N$ 又は $ABR^{-1} \bmod N$ の演算結果 W を第 1 の記憶回路に格納し、演算器のオーバーフローフラグ OV の有無を記憶し、上記剰余乗算の後に上記第 1 の記憶回路と第 2 の記憶回路のアドレスを交換し、第 2 の記憶回路を選択するアドレスにより選択される演算結果値 W から N の減算 $W - N$ を行って第 1 の記憶回路を選択するアドレスにより選択される第 2 の記憶回路に減算結果 $W - N$ を格納し、オーバーフローフラグ OV が有るときにのみ上記アドレスを再度交換し、第 1 の記憶回路を選択するアドレスにより選択される第 1 又は第 2 の記憶回路のデータを有効なデータとして出力させることにより、暗号化処理用演算ユニットの簡素化に加えてレジスタへの書き込み論理の簡素化と高速化を図りつつ電流波形を利用した解読を困難にすることができるという効果が得られる。

【 0 1 2 0 】

(31) 上記に加えて、上記マイクロコンピュータの暗号化処理用演算ユニットにおいて、上記 $A^2 R^{-1} \bmod N$ 又は $ABR^{-1} \bmod N$ の演算結果 W を第 1 の記憶回路に格納し、上記剰余乗算の後に上記第 1 の記憶回路の演算結果値 W から N の減算 $W - N$ を行って第 2 の記憶回路に格納し、この $W - N$ の減算が行われた時の演算器からボロフラグ BR を記憶し、ボロフラグ BR が有るときには、第 1 の記憶回路と第 2 の記憶回路を選択する最下位アドレスを逆転させて、上記第 2 の記憶回路を選択するアドレスにより第 1 の記憶回路の演算結果 W を出力し、ボロフラグ BR が無いときには、第 1 の記憶回路と第 2 の記憶回路を選択する最下位アドレスをそのままして、上記第 2 の記憶回路を選択するアドレスにより第 2 の記憶回路の演算結果 $W - N$ を出力させることにより、暗号化処理用演算ユニットの簡素化に加えてレジスタへの書き込み論理の簡素化と高速化を図りつつ電流波形を利用した解読を困難にすることができるという効果が得られる。

【 0 1 2 1 】

以上本発明者よりなされた発明を実施例に基づき具体的に説明したが、本願発明は前記実施例に限定されるものではなく、その要旨を逸脱しない範囲で種々変更可能であることはいうまでもない。例えば、ＩＣカードには、１つの半導体集積回路装置を搭載するもの他、複数の半導体集積回路装置が搭載されるものであってもよい。マイクロコンピュータは、１つの半導体集積回路装置に形成されるもの他、ＣＰＵとその周辺回路が複数チップで構成されて、１つのモジュール基板に搭載されてなるものであってもよい。

【 0 1 2 2 】

演算処理は前記のような暗号処理を行なうべき乗剰余乗算法の他に、図 2 5 図に示したフローチャート図のように演算 A と演算 B を持ち、演算 A の結果により演算 B を行なうか否かの分岐を持つような演算処理、あるいは演算動作でのオーバーフローの有無に対応して、次の演算処理が選択的に追加される場合に等に広く利用することができる。つまり、演算 A の次に演算 B を実行し、演算 A の結果から演算 B が不要なら、その演算結果を無効にするような演算処理を行なえば、前記のような暗号処理以外の機密動作を必要とするデータ処理のハッキング対策として有益なものとなる。

【 0 1 2 3 】

上記マイクロコンピュータは、データ処理装置とかかるデータ処理装置によるデータ処理手順が書き込まれた ROM を含んで記データ処理手順に従ってデータの入出力動作が行われるものであれば何であってもよい。例えば、前記のようなＩＣカード用チップの他に、ゲーム用等の１チップマイクロコンピュータ等のように機密保護の必要な各種マイクロコンピュータに広く適用できるものである。この発明は、機密保護を必要とする各種ＩＣカード及びマイクロコンピュータに広く利用できる。

【 0 1 2 4 】

【発明の効果】

本願において開示される発明のうち代表的なものによって得られる効果を簡単に説明すれば、下記の通りである。すなわち、外部端子がリードライト装置と電気的に接続されることによって動作電圧が供給され、かつ、暗号化処理又は復号

化処理を伴ったデータの入出力動作を含む I C カードにおいて、上記暗号化処理又は復号化処理に攪乱目的のダミー処理動作を含ませて内部回路の動作タイミング及び動作電流の画一化を行なうようにすることによって、電流波形を利用した解読を困難にすることができる。

【 0 1 2 5 】

暗号化処理又は復号化処理を伴ったデータの入出力動作を含むモジュール構成のマイクロコンピュータにおいて、上記暗号化処理又は復号化処理に攪乱目的のダミー処理動作を含ませて内部回路の動作タイミング及び動作電流の画一化を行なうようにすることにより、モジュール化されたマイクロコンピュータに対する電流波形を利用した解読を困難にすることができる。

【図面の簡単な説明】

【図 1】

この発明が適用される I C カードの一実施例を示す外観図である。

【図 2】

この発明に係る I C カードに搭載される I C カード用チップの一実施例を示す概略ブロック図である。

【図 3】

この発明に係るコプロセッサの一実施例の動作を説明するためのタイミング図である。

【図 4】

図 3 のコプロセッサの動作を説明するためのフローチャート図である。

【図 5】

図 3 のコプロセッサの一実施例を示すブロック図である。

【図 6】

図 3 に示したコプロセッサの動作を実現するための一実施例を示すブロック図である。

【図 7】

図 3 のコプロセッサの他の一実施例を示すブロック図である。

【図 8】

図 3 のコプロセッサの他の一実施例を示すブロック図である。

【図 9】

この発明に係るコプロセッサの他の一実施例の動作を説明するための構成図である。

【図 1 0】

図 9 に示したコプロセッサの動作を実現するための一実施例を示すブロック図である。

【図 1 1】

この発明に係るコプロセッサの他の一実施例の動作を説明するためのタイミング図である。

【図 1 2】

この発明に係るコプロセッサの他の一実施例の動作を説明するためのフローチャート図である。

【図 1 3】

この発明に係るコプロセッサの他の一実施例の動作の詳細を説明するためのタイミング図である。

【図 1 4】

図 1 1 ないし図 1 3 に示したコプロセッサの動作を実現するための一実施例を示すブロック図である。

【図 1 5】

この発明に係るコプロセッサの更に他の一実施例の動作を説明するためのタイミング図である。

【図 1 6】

この発明に係るコプロセッサの演算動作の他の一実施例を示すフローチャート図である。

【図 1 7】

この発明に係るコプロセッサの他の一実施例を示すブロック図である。

【図 1 8】

この発明に係るコプロセッサの他の一実施例を示すブロック図である。

【図 1 9】

この発明に係るコプロセッサの他の一実施例を示すブロック図である。

【図 2 0】

この発明に係る IC カード用チップの他の一実施例を示す要部ブロック図である。

【図 2 1】

図 2 0 のカウンタの一実施例を示すブロック図である。

【図 2 2】

図 2 0 の IC カード用チップの動作の一例を示すタイミング図である。

【図 2 3】

この発明に係る IC カード用チップの更に他の一実施例を示す要部ブロック図である。

【図 2 4】

図 2 3 の IC カード用チップの動作の一例を示すタイミング図である。

【図 2 5】

この発明が適用可能な演算動作を説明するためのフローチャート図である。

【図 2 6】

この発明に用いられるコプロセッサの他の一実施例を示すブロック図である。

【図 2 7】

この発明における「 $R^2 \bmod N$ 」の計算方法を示す概念図である。

【図 2 8】

この発明に係る暗号化処理用演算ユニットの一実施例を示す要部ブロック図である。

【図 2 9】

この発明に係る暗号化処理用演算ユニットの他の一実施例を示す要部ブロック図である。

【図 3 0】

この発明に係る暗号化処理用演算ユニットの更に他の一実施例を示す要部ブロック図である。

【符号の説明】

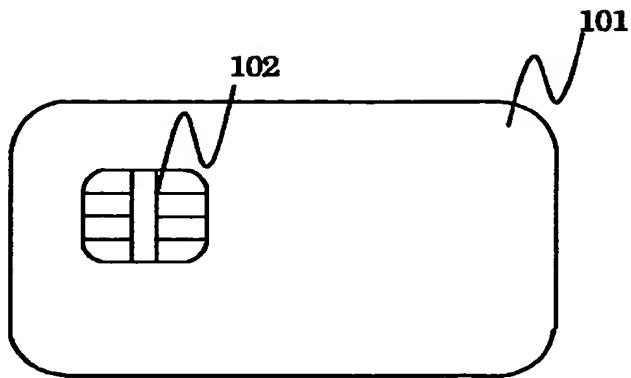
2 0 1 … 中央処理装置 (C P U)、2 0 2 … I / O ポート、2 0 3 … アドレスバス、2 0 4 … データバス、2 0 5 … クロック生成回路、2 0 6 … R O M、2 0 7 … R A M、2 0 8 … E E P R O M、2 0 9 … コプロセッサ (暗号化処理用演算ユニット)、

C D A、C D B、C D N、C D W … レジスタ。

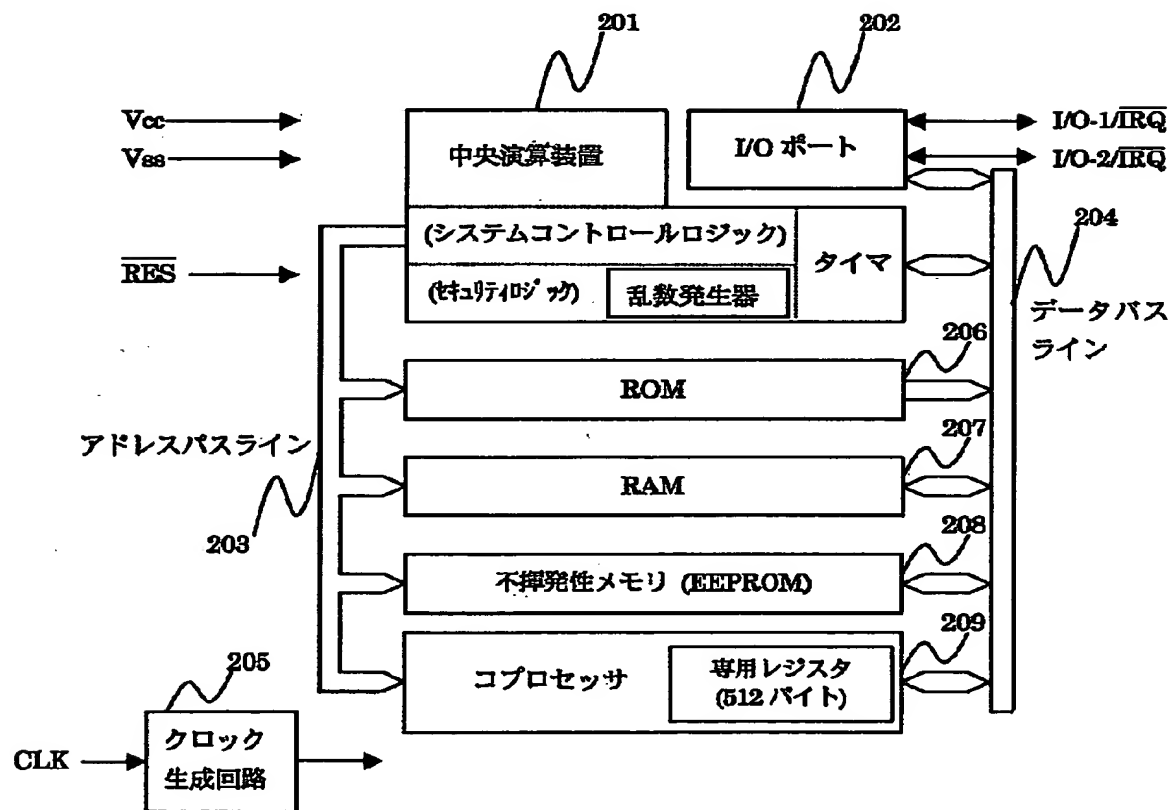
3 3、3 4 … 積和演算器、3 5 … テンポラリレジスタ、3 6 ~ 3 8 … レジスタ、3 9 … M i 生成ロジック、4 0 … M i を保持するラッチ (レジスタ)、4 1 … シフタ、4 2 … R A M、4 3 … データバス。

【書類名】 図面

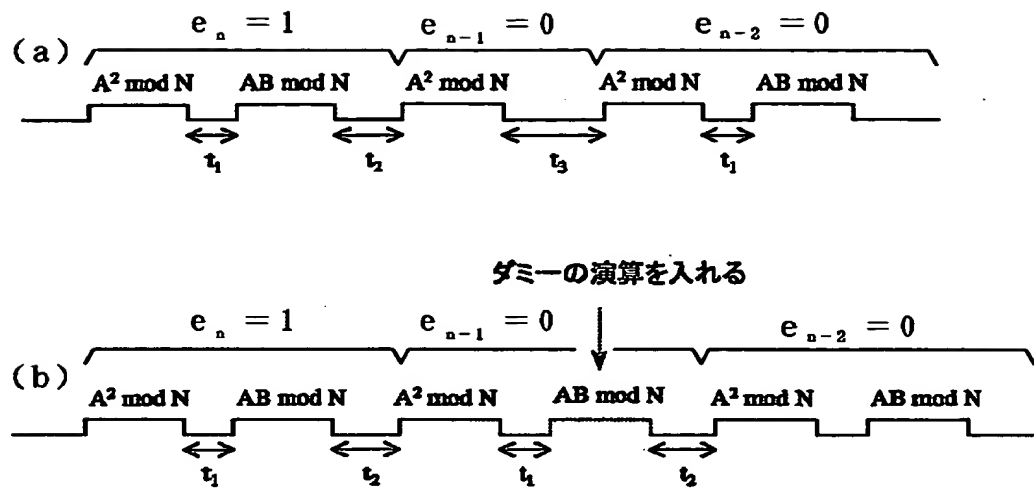
【図 1】



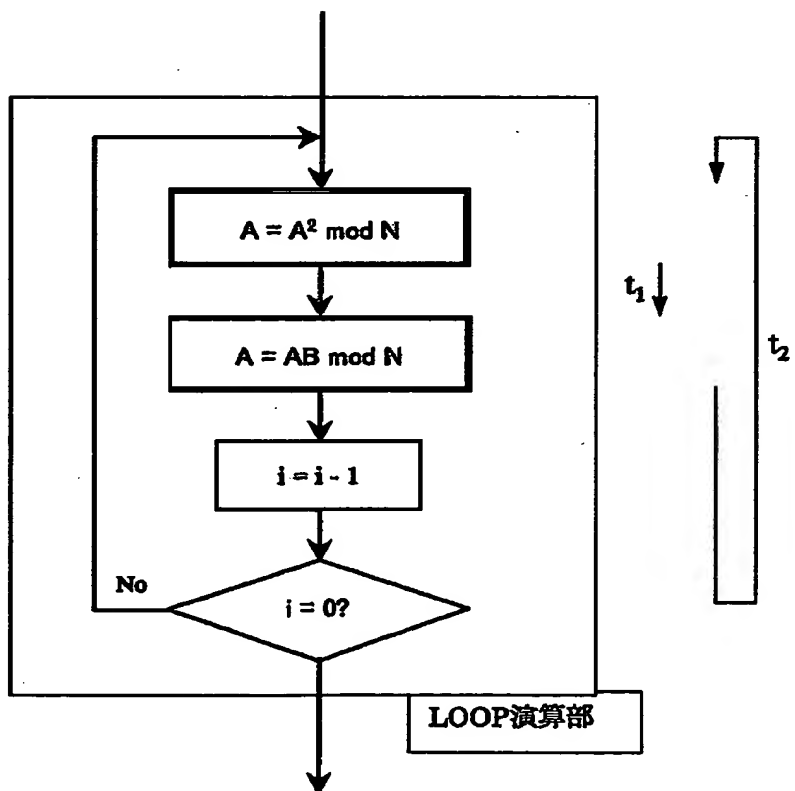
【図 2】



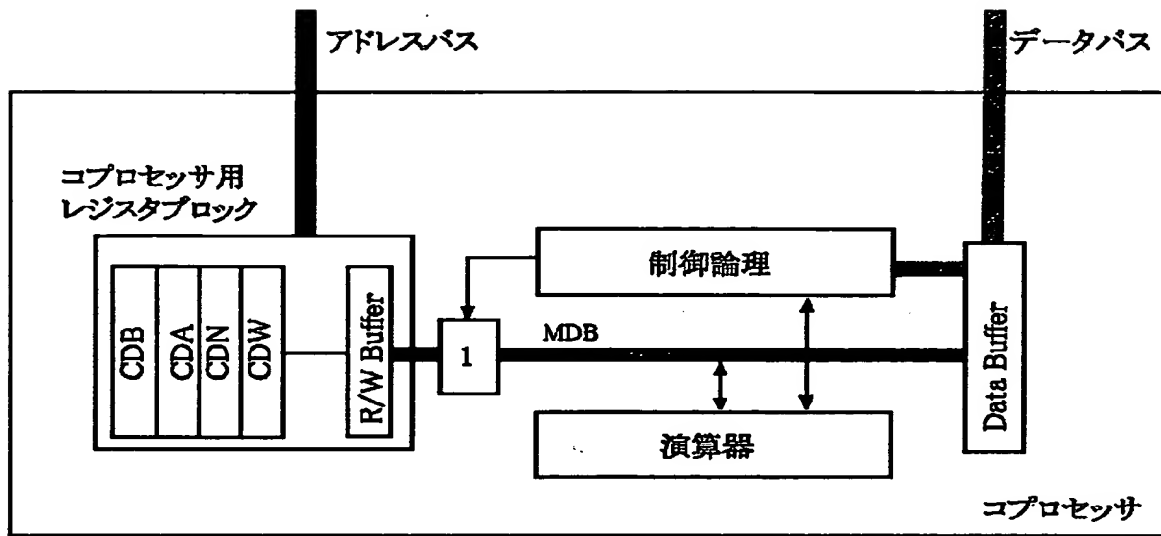
【図 3】



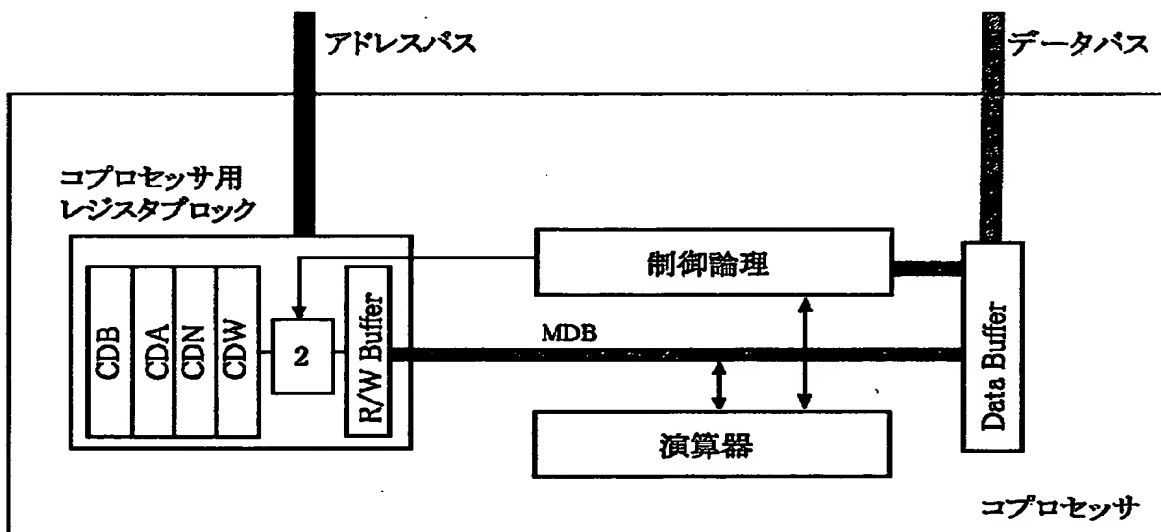
【図 4】



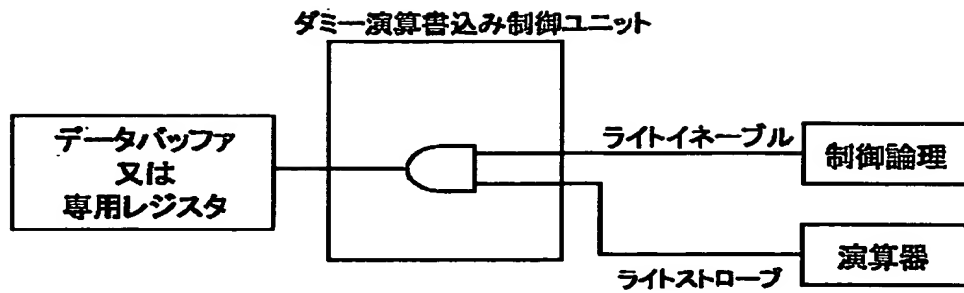
【図 5】



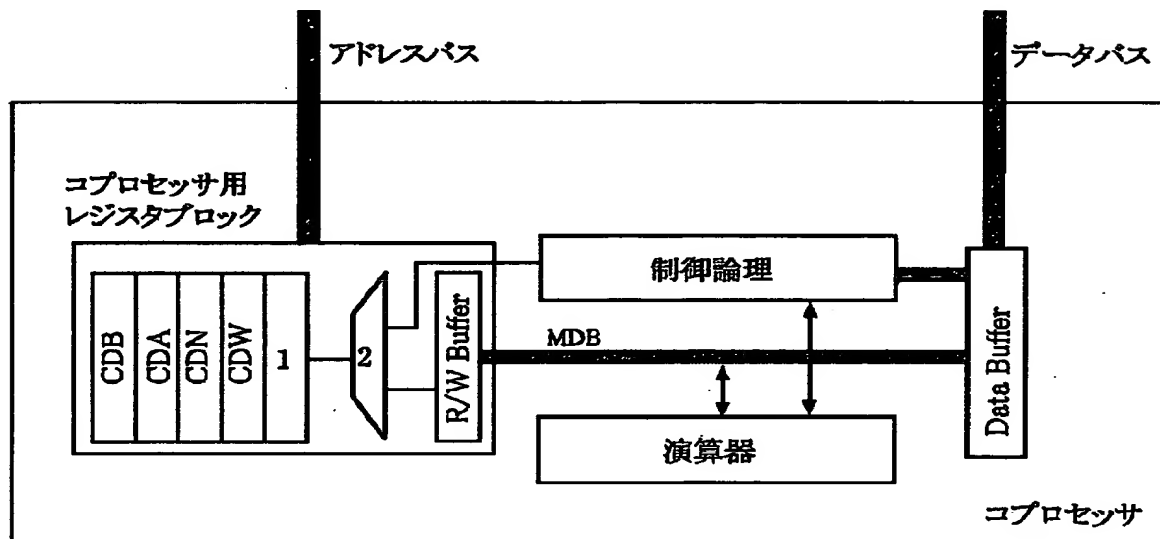
【図 6】



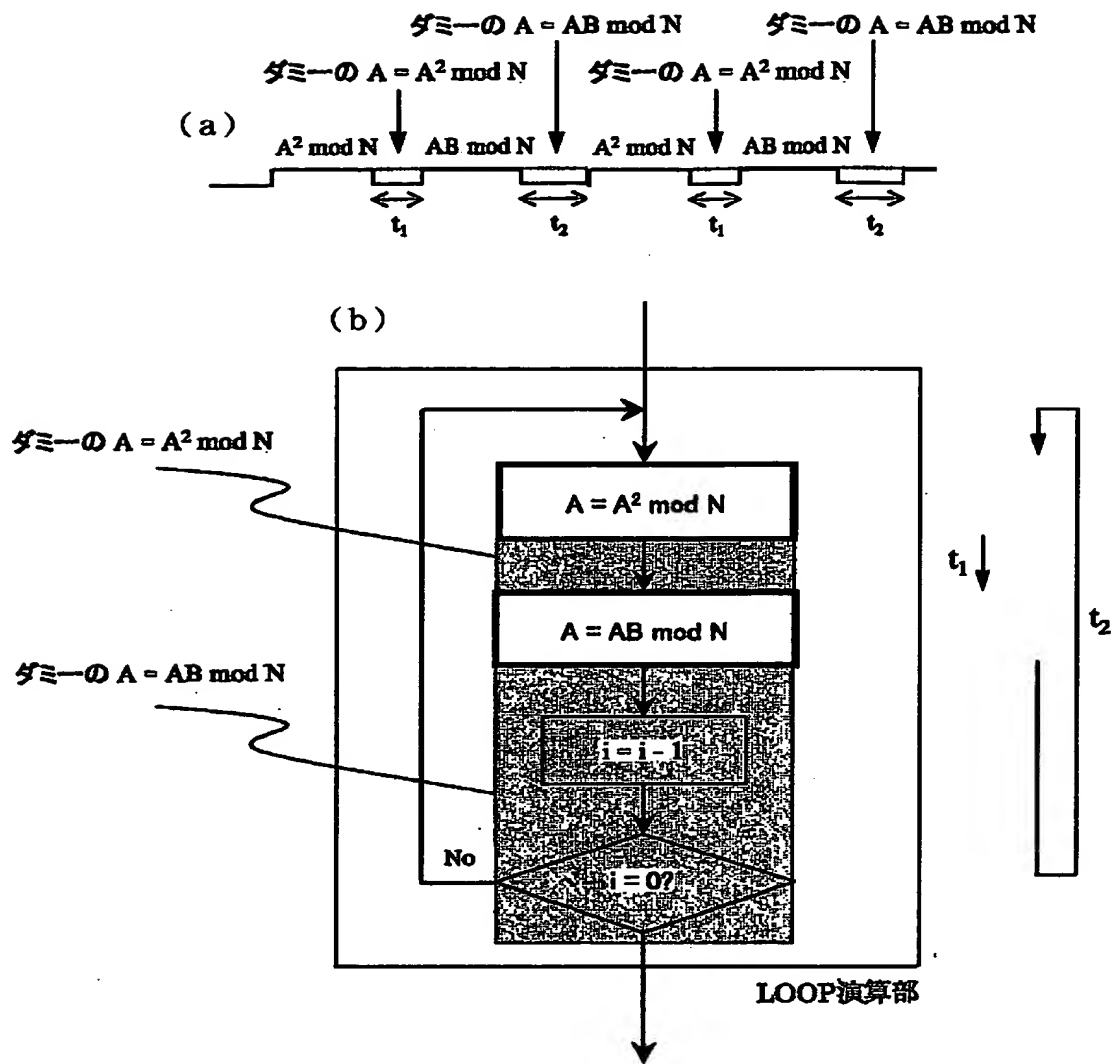
【図 7】



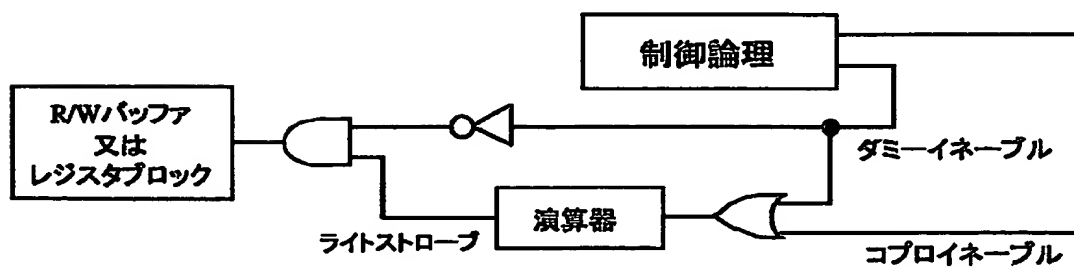
【図 8】



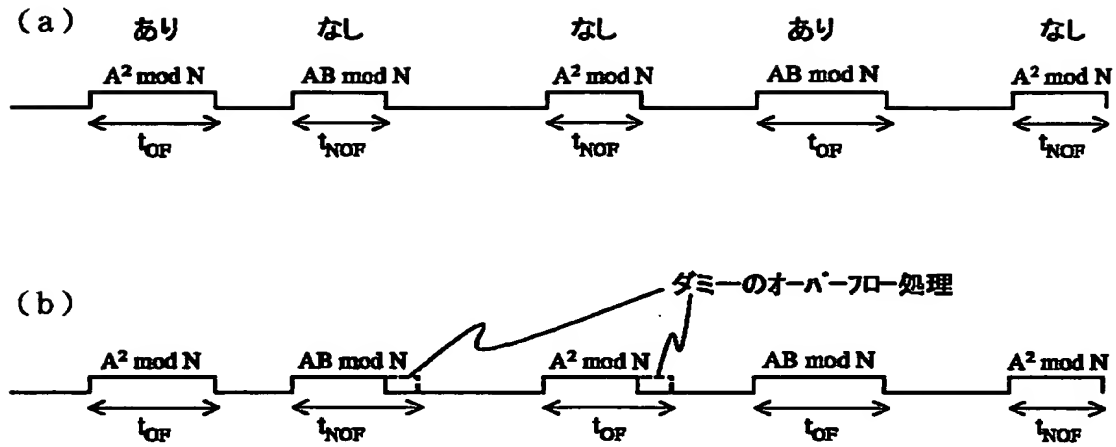
【図 9】



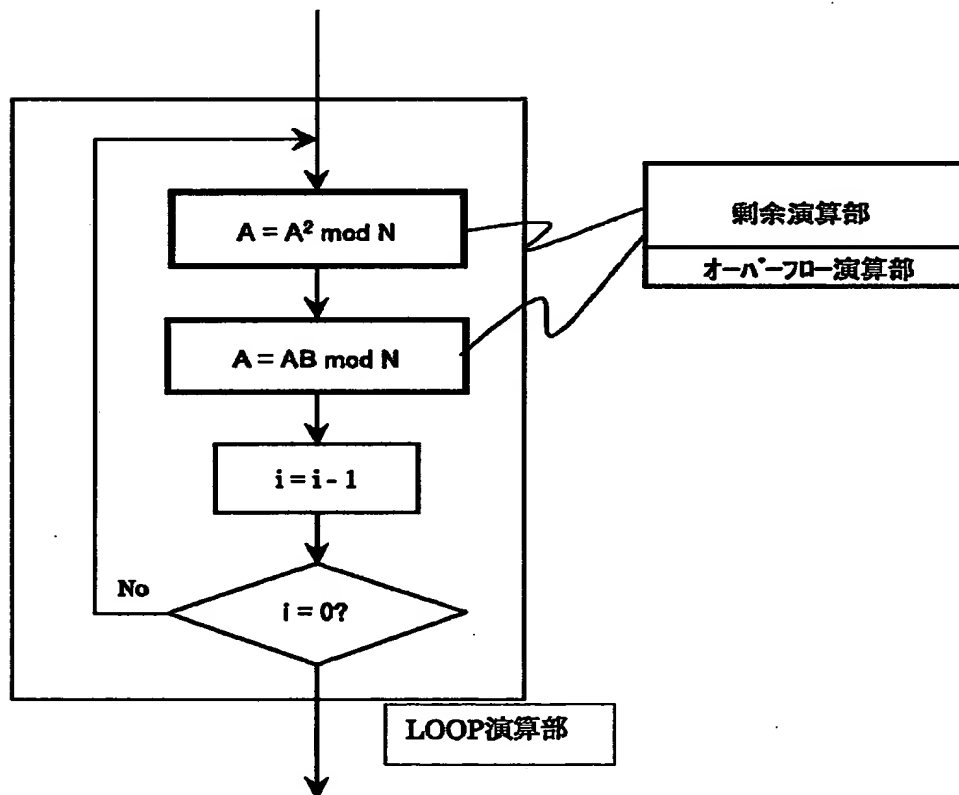
【図 10】



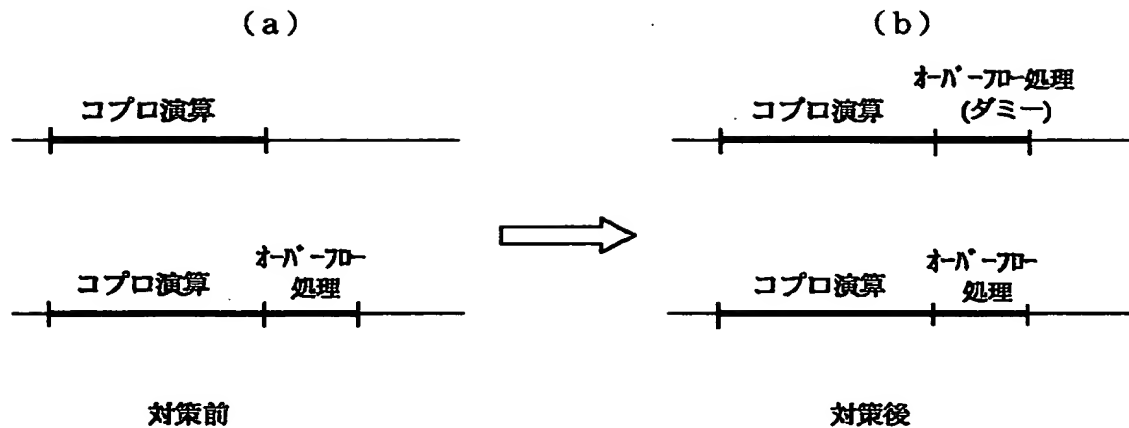
【図 1 1】



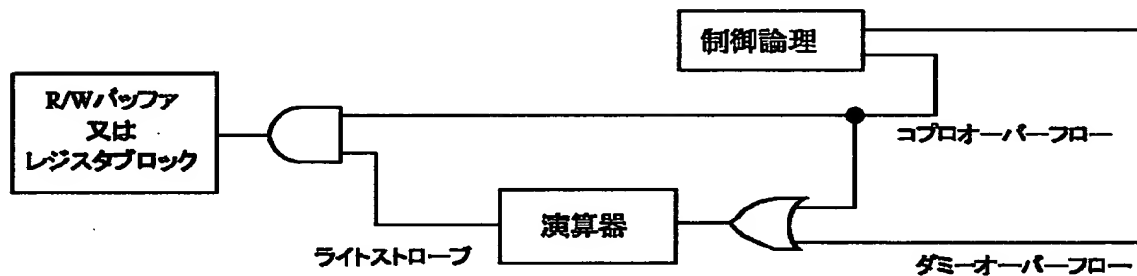
【図 1 2】



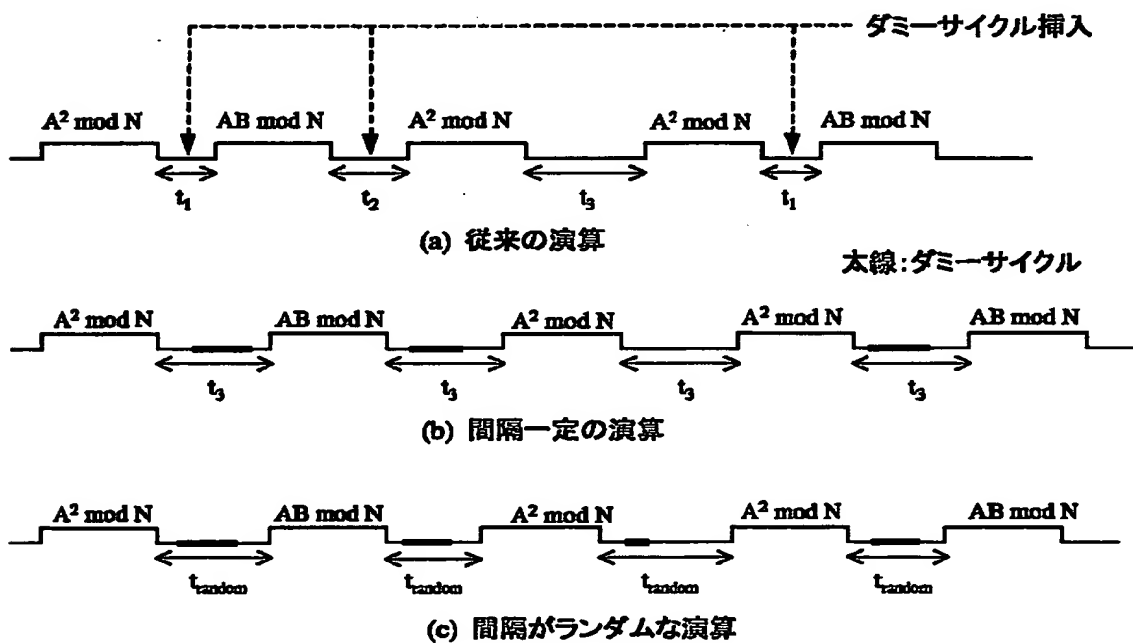
【図13】



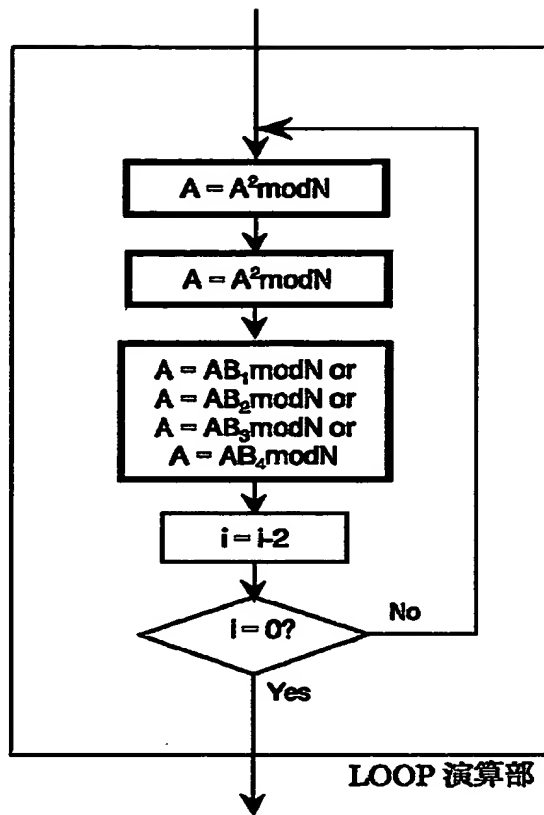
【図14】



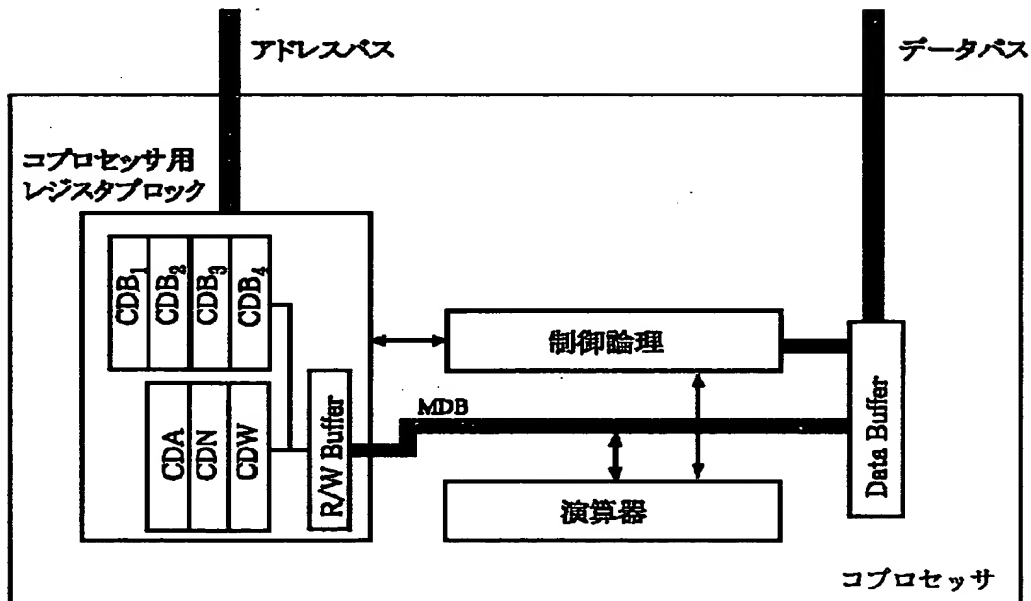
【図15】



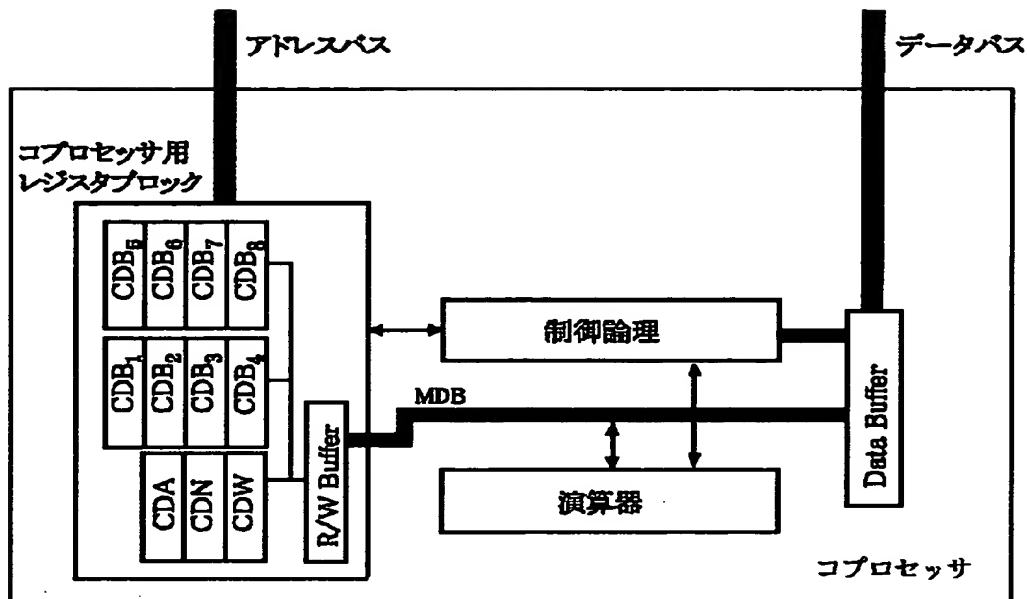
【図 16】



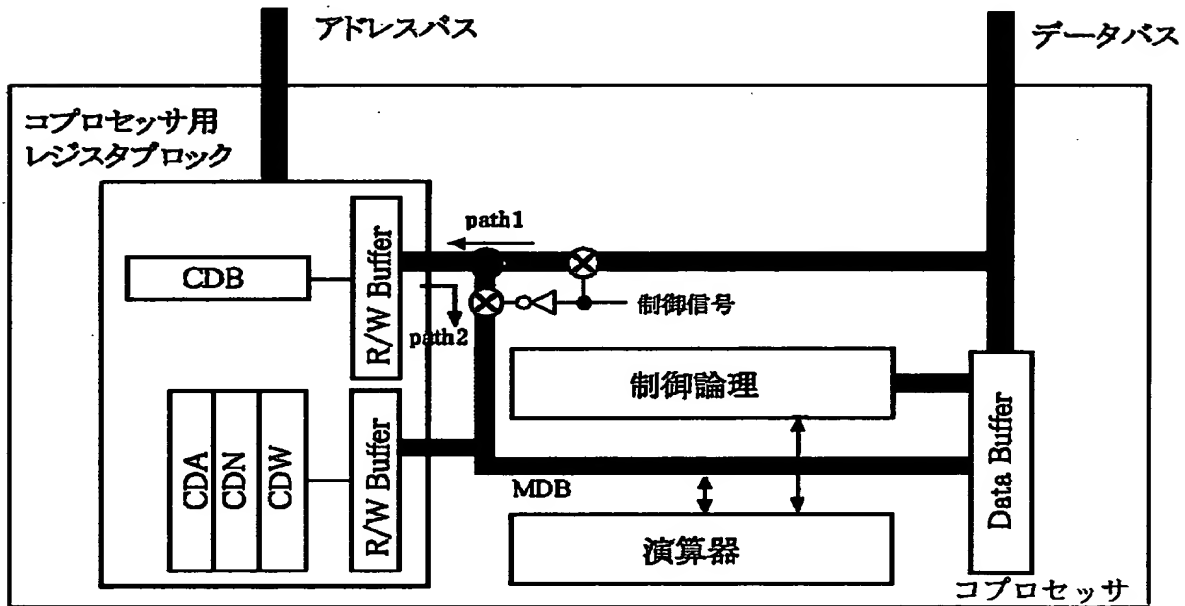
【図 17】



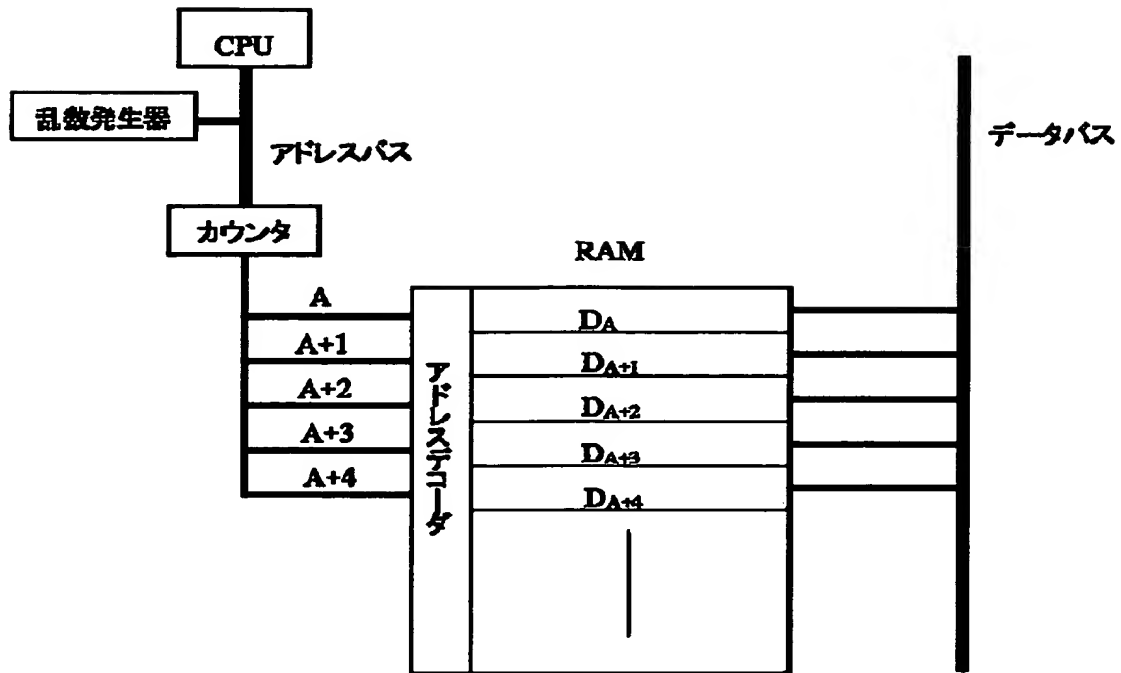
【図 18】



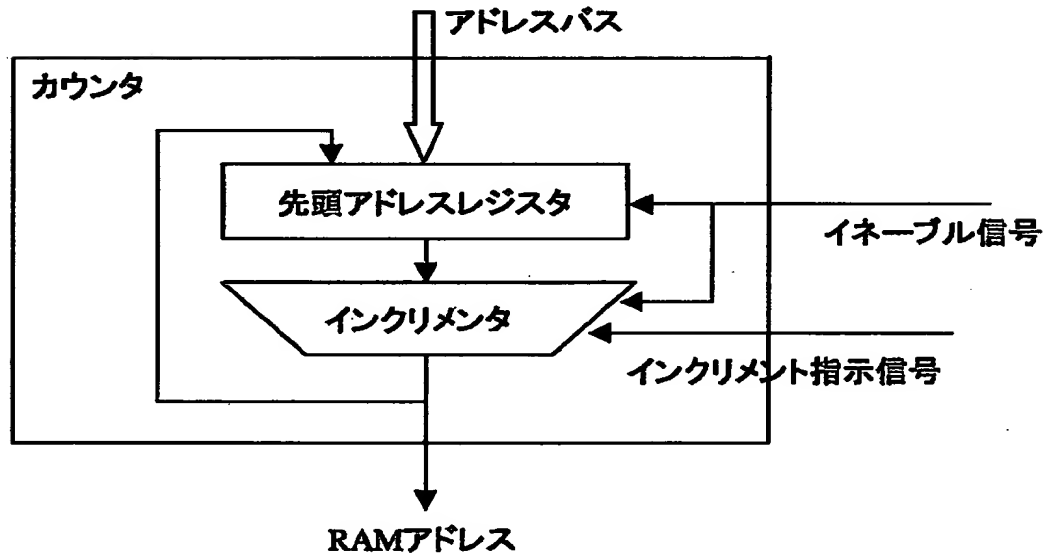
【図 19】



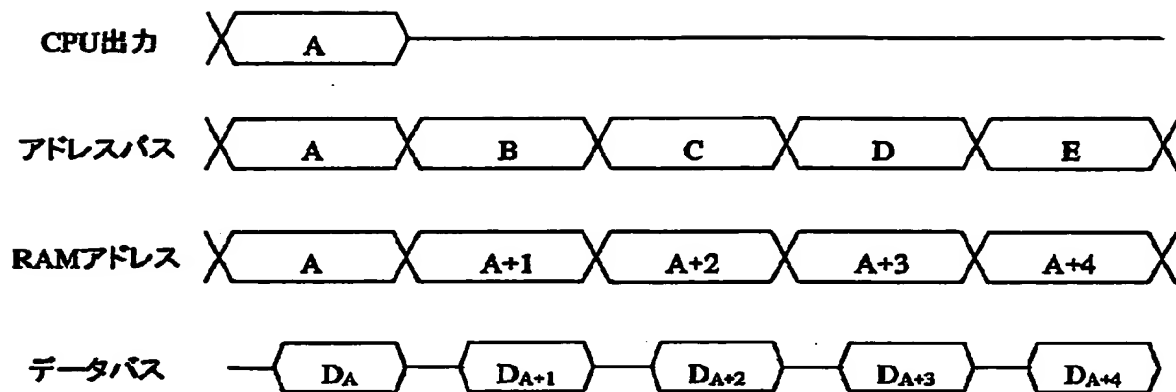
【図 2 0】



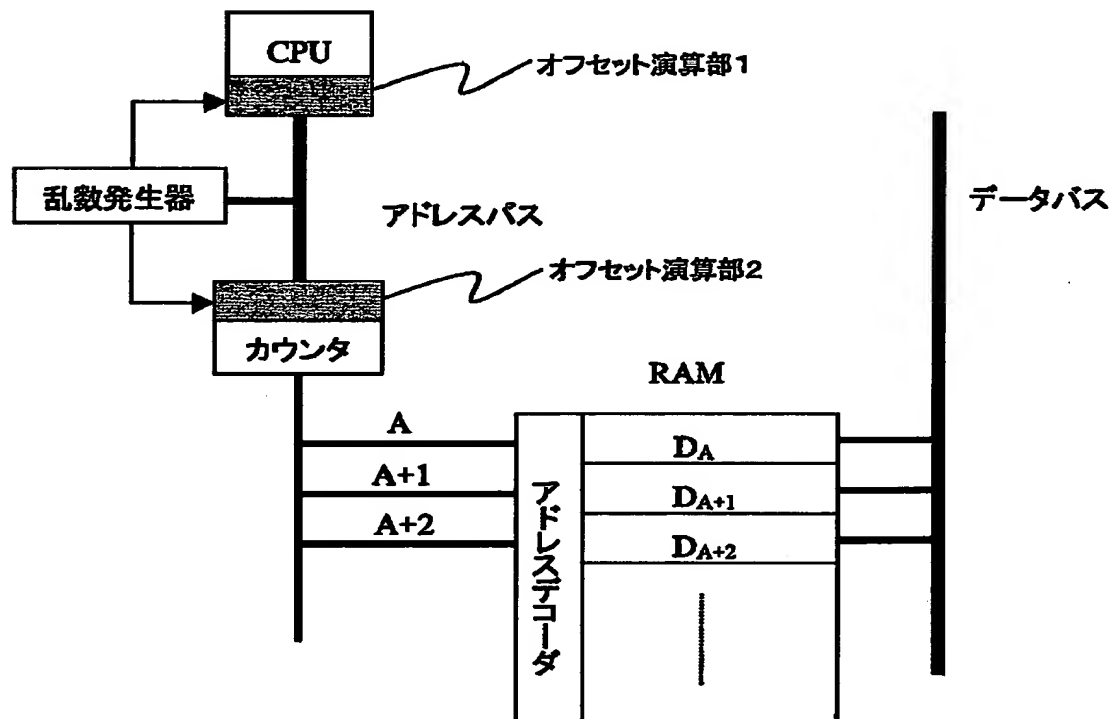
【図 2 1】



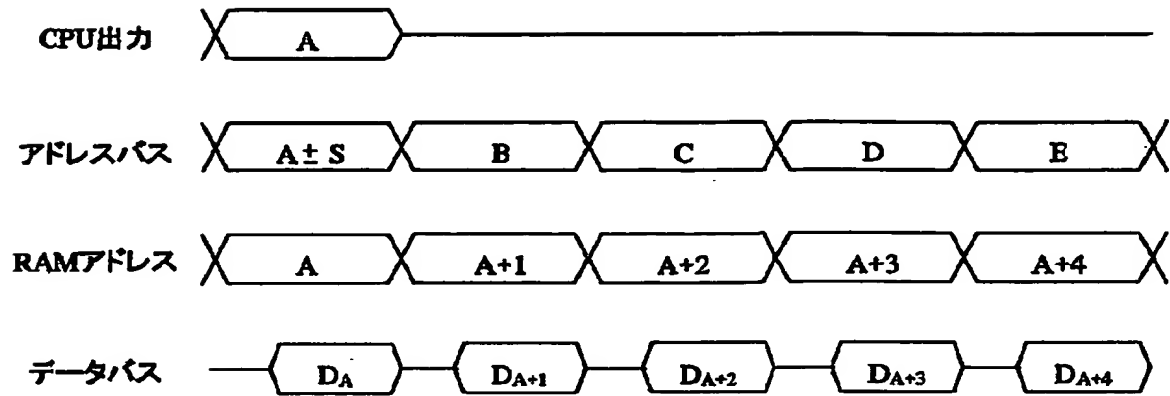
【図 2 2】



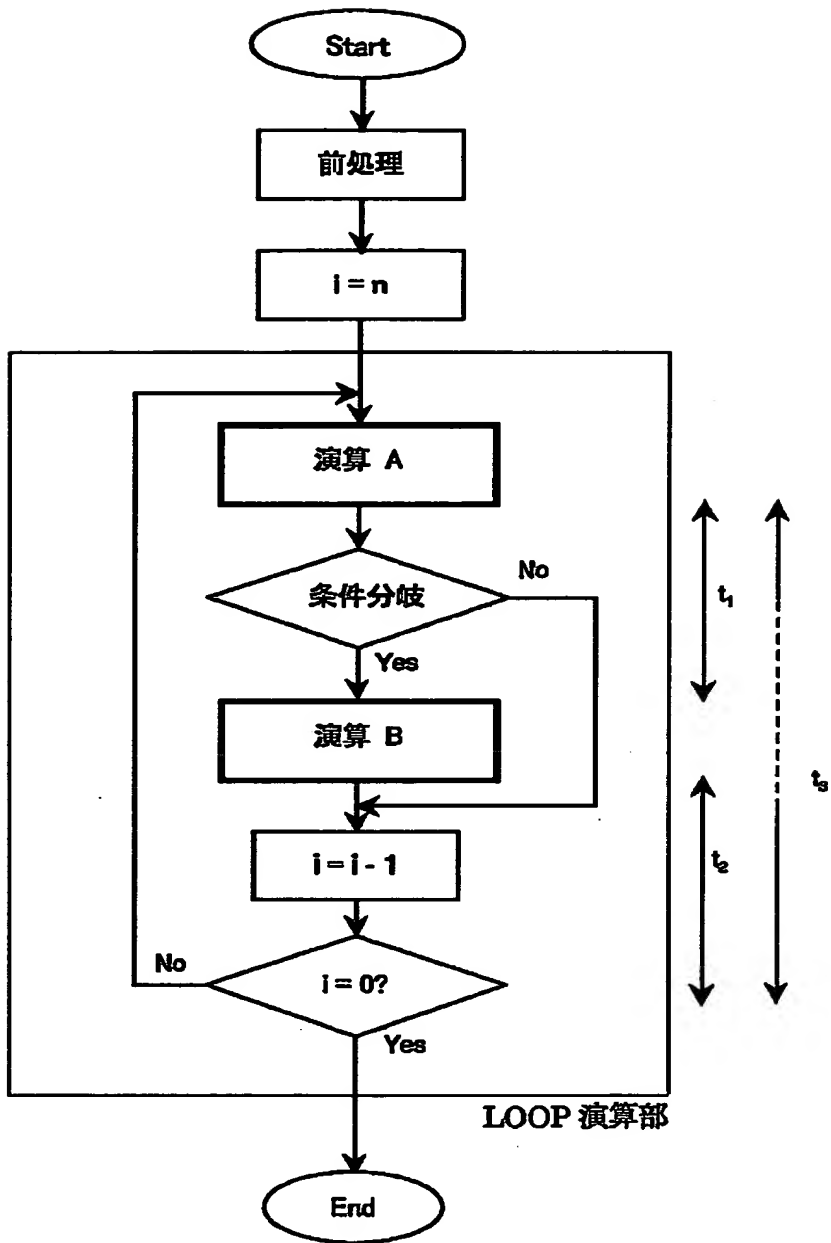
【図 2 3】



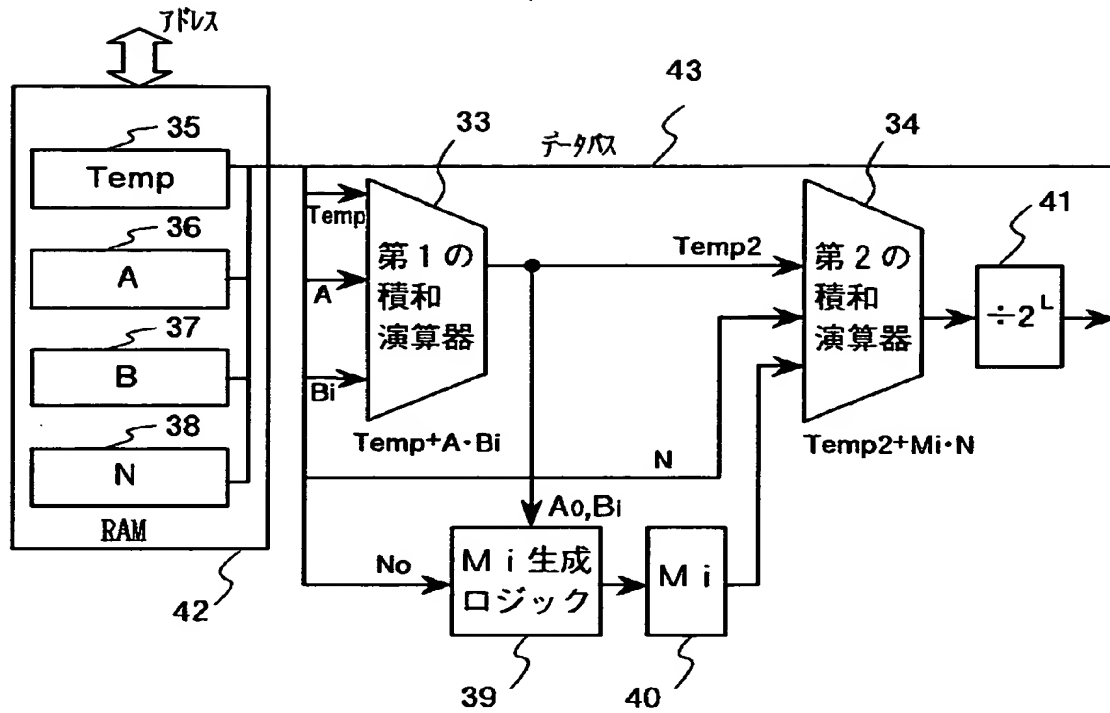
【図 2 4】



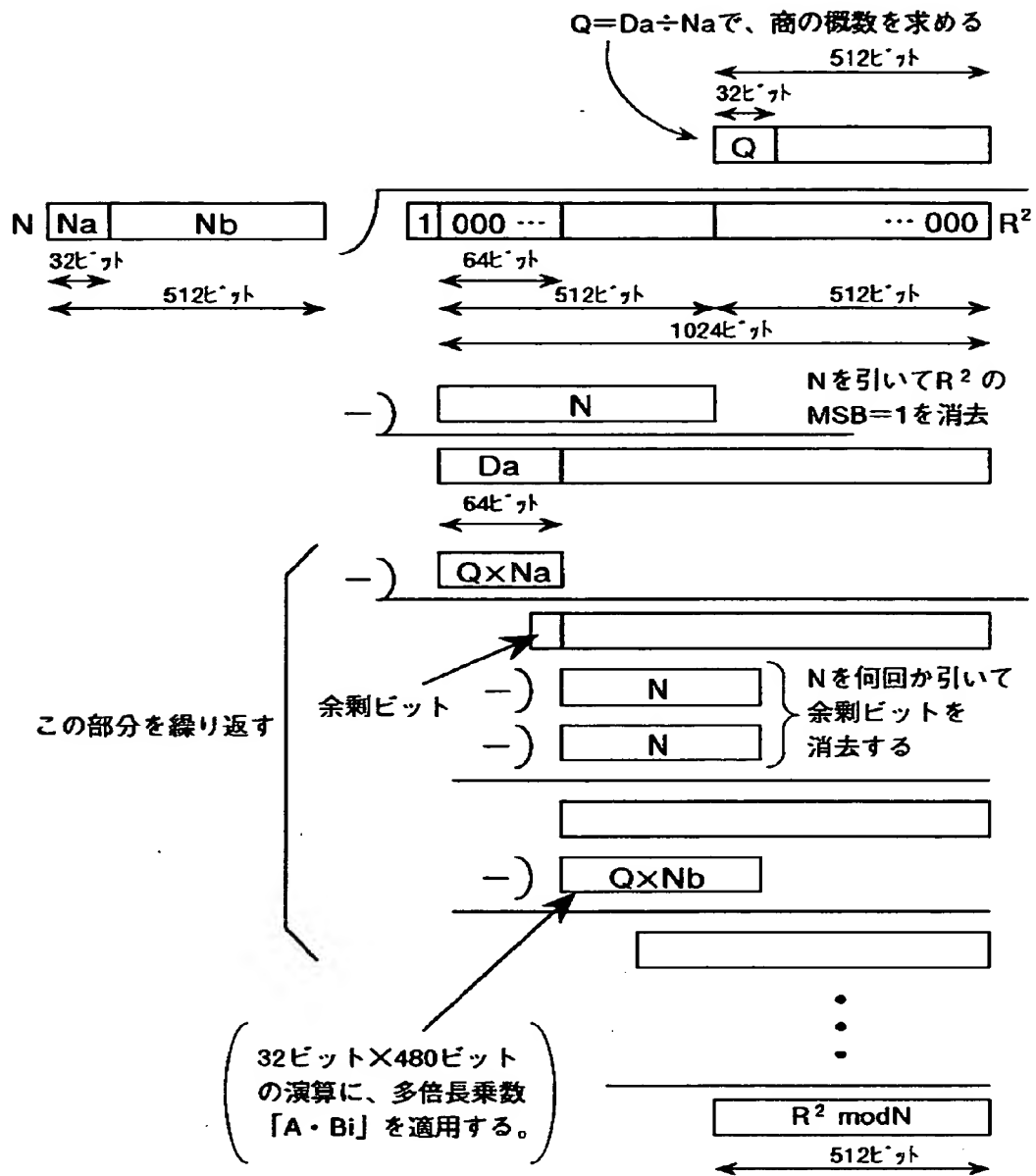
【図 2 5】



【図 26】

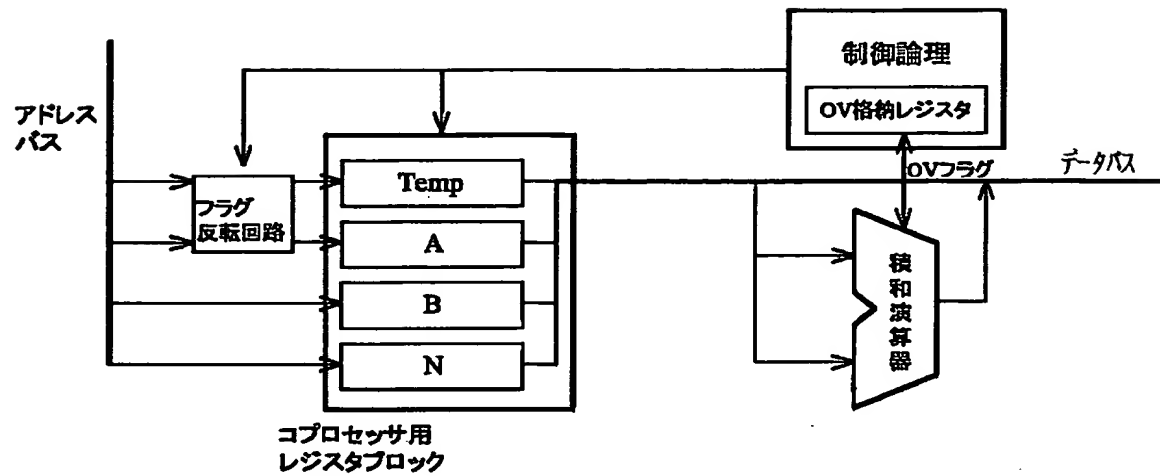


【図 2 7】

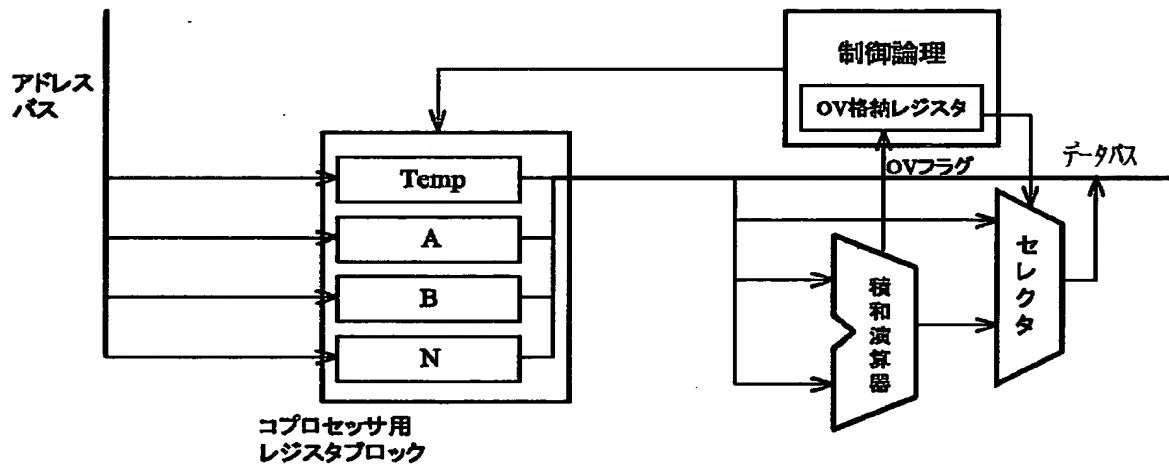


$R^2 \bmod N$ の計算概念図

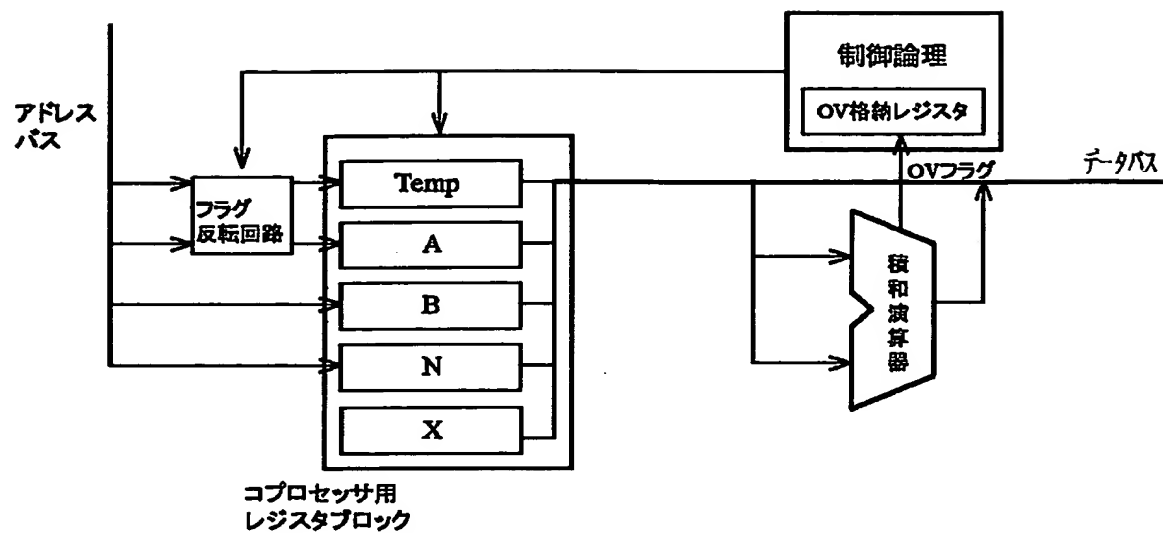
【図 28】



【図 29】



【図 3 0】



【書類名】 要約書

【要約】

【課題】 機密保護の強化及び機密保護のための信号処理の高速化とその強化を実現したＩＣカードとマイクロコンピュータを提供する。

【解決手段】 外部端子がリードライト装置と電氣的に接続されることによって動作電圧が供給され、かつ、暗号化処理又は復号化処理を伴ったデータの入出力動作を含むＩＣカードにおいて、上記暗号化処理又は復号化処理に攪乱目的のダミー処理動作を含ませて内部回路の動作タイミング及び動作電流の画一化を行なうようにする。暗号化処理又は復号化処理を伴ったデータの入出力動作を含むモジュール構成のマイクロコンピュータにおいて、上記暗号化処理又は復号化処理に攪乱目的のダミー処理動作を含ませて内部回路の動作タイミング及び動作電流の画一化を行なうようにする。

【選択図】 図 4

出 願 人 履 歴 情 報

識別番号 [000005108]

1. 変更年月日 1990年 8月31日
[変更理由] 新規登録
住 所 東京都千代田区神田駿河台4丁目6番地
氏 名 株式会社日立製作所

出 願 人 履 歴 情 報

識別番号 [000233169]

1. 変更年月日	1998年 4月 3日
[変更理由]	名称変更
住 所	東京都小平市上水本町5丁目22番1号
氏 名	株式会社日立超エル・エス・アイ・システムズ